11/40/45

INSTRUCTION EXERCISER

MD-11-DCQKC-G

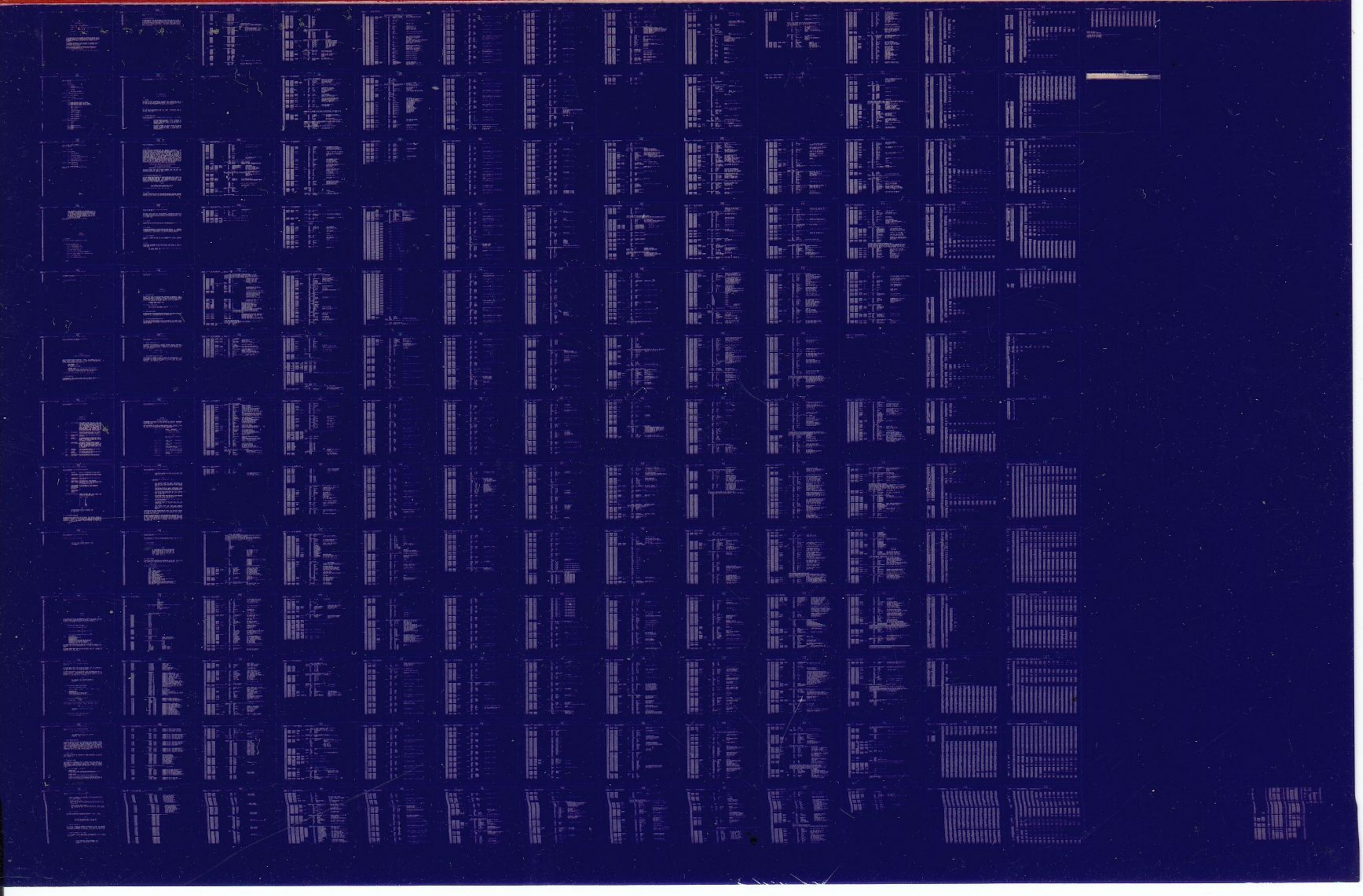
EP-DCQKC-G-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



.REM !

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCQKC-G

PRODUCT NAME: 11/40 AND 11/45 INSTRUCTION EXERCISER

DATE: MAY,1976

MAINTAINER: DIAGNOSTIC GROUP

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
TABLE OF CONTENTS

ABSTRACT

CHAPTER 1 REQUIREMENTS

1.1 EQUIPMENT
1.1.2 OPTIONAL EQUIPMENT USED

1.2 STORAGE

1.3 PRELIMINARY PROGRAMS

CHAPTER 2 LOADING AND STARTING PROCEDURE

2.1 ACT11 OPERATION

CHAPTER 3 SWITCH SETTINGS

3.1 11/45 DISPLAY REGISTER

CHAPTER 4 ERRORS

4.1.1 ERROR PRINTOUT FORMAT (CP ERROR)
4.1.2 ERROR PRINTOUT FORMAT (DEVICE ERROR)
4.1.3 ERROR PRINTOUT FORMAT (PARITY ERROR)
4.1.4 ERROR PRINTOUT FORMAT (RELOCATION ERROR)

4.2 PARITY ERROR DETECTION

4.3 ERROR LOOPING

4.4 UNPREDICTED ERRORS

4.5 TRAP TO LOCATION 4

4.6 TRAP TO LOCATION 10

4.7 MEMORY MANAGEMENT (KT11) ABORT

4.8 ERROR DISCUSSION

CHAPTER 5 SUBROUTINE ABSTRACTS

5.1 SCOPEA

5.2 ERROR

5.3 PROGRAM RELOCATION
5.3.1 RELOC
5.3.2 RELOCATION ABOVE 28K (STMM)
5.3.3 IODEV
5.3.4 WAITIO

5.4 CLOCK INTERRUPT
PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
TABLE OF CONTENTS

5.5 END

CHAPTER 6 MISCELLANEOUS

6.1 EXECUTION TIME

6.2 PASS MODIFICATION

6.3 I/O DEVICE ADDRESS MODIFICATION

6.4 MEMORY MODIFICATION

6.5 USER DEFINED RELOCATION LIMITS

CHAPTER 7 PROGRAM DESCRIPTION

7.1 STACK POINTER

7.2 POWER FAILURE
PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
ABSTRACT

THIS DIAGNOSTIC PROGRAM IS DESIGNED TO BE A COMPREHENSIVE CHECK OF THE PDP-11/40 AND PDP-11/45 PROCESSORS. THE PROGRAM EXECUTES EACH INSTRUCTION IN ALL ADDRESS MODES AND INCLUDES TESTS FOR TRAPS AND THE TELETYPE INTERRUPT SEQUENCE. THE PROGRAM RELOCATES THE TEST CODE THROUGHOUT MEMORY 0-124K. IF SELECTED, THE PROGRAM MAY BE RELOCATED BY ANY OF THE AVAILABLE DISKS.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
REQUIREMENTS

CHAPTER 1
REQUIREMENTS

1.1 EQUIPMENT

PDP-11 FAMILY CENTRAL PROCESSOR WITH 8K MEMORY.

1.1.2 OPTIONAL EQUIPMENT USED

1. KW11-P (PROGRAMMABLE CLOCK)
2. KW11-L (LINE FREQUENCY CLOCK)
3. ALL PARITY MEMORY OPTIONS
4. KT11-C,D (11/40, 11/45 MEMORY MANAGEMENT)
5. RK11, RF11, RP11, RS03/4, RC11, RP04/05/06,RK06
6. KJ-11 (11/40 STACK LIMIT)
7. EIS (11/40 EXTENDED INSTRUCTION SET)

1.2 STORAGE

THE PROGRAM LOADS INTO THE FIRST 6K OF MEMORY, AND RUNS IN ALL MEMORY
(EXCLUSIVE OF LOADERS).

1.3 PRELIMINARY PROGRAMS

NONE.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
LOADING AND STARTING PROCEDURE

CHAPTER 2

LOADING AND STARTING PROCEDURE

LOAD THE PROGRAM USING THE ABSOLUTE LOADER. IF CONSOLE TTY IS A SERIAL DEVICE (LA30S, VT05, ETC.), FILLER CHARACTERS ARE REQUIRED. DEPOSIT INTO LOCATION 1002 (FILLS) A '0 (THE FILLER CHARACTER) AND LOCATION 1003 11(OCTAL) (THE FILLER COUNT).

LOAD ADDRESS = 200
PRESS START,
SET OPERATING SWITCHES

CONTENTS OF OPT.CP IS TYPED ON FIRST PASS (SEE CHAPT 7)
(INITIAL LOAD)
PASS COUNT IS PRINTED AFTER EACH PASS (SEE SECTION 5.5)
"DCQKC DONE" IS PRINTED WHEN DONE (SEE SECTION 6.1).

IF NO CONSOLE TTY IS AVAILABLE, SET SW15=1 (HALT ON ERROR).

2.1 ACT11 OPERATION

IF THE PROGRAM IS RUN IN QUICK VERIFY MODE, NO SUBTEST ITERATIONS ARE PERFORMED BUT ALL AVAILABLE DISKS ARE RUN ROUND ROBIN.(SEE SECTION 3.0, SW05.)

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
SWITCH SETTINGS

CHAPTER 3

SWITCH SETTINGS

SW15	HALT ON ERROR	THIS SWITCH WHEN SET WILL HALT THE PROCESSOR WHEN AN ERROR IS DETECTED. THE PC+2 AND THE CURRENT STATUS AT THE TIME OF THE ERROR IS STORED ON THE STACK (R6). IF THIS SWITCH IS SET BEFORE AN ERROR IS DETECTED, THE PROGRAM HALTS AS DESCRIBED ABOVE. THE PROGRAM MAY BE HALTED AFTER THE ERROR TYPEOUT OCCURS BY SETTING SW15 AFTER THE TYPEOUT BEGINS.
SW14	LOOP SUBTEST	THIS SWITCH WHEN SET LOOPS THE CURRENT SUBTEST RUNNING REGARDLESS OF ERROR.
SW13	INHIBIT ERROR PRINTOUT	THIS SWITCH WHEN SET INHIBITS THE ERROR PRINTOUT.
SW12	INHIBIT RELOCATION	THIS SWITCH WHEN SET CAUSES THE PROGRAM TO BE EXECUTED ONLY IN THE FIRST 8K OF MEMORY. THIS SWITCH CANNOT BE SET WHEN THE PROGRAM IS RUNNING.
SW11	INHIBIT SUB-TEST ITERATION	THIS SWITCH WHEN SET INHIBITS SUBTEST REITERATION. NORMALLY EACH SUBTEST IS EXECUTED 8 TIMES BEFORE THE NEXT SUBTEST IS RUN. SETTING SW11 CAUSES EACH TEST TO BE EXECUTED ONCE BEFORE STARTING THE NEXT SUBTEST.
SW10	RING BELL ON ERROR	THIS SWITCH WHEN SET WILL RING THE BELL WHEN AN ERROR IS DETECTED.
SW9	INHIBIT RELOCATION	THIS SWITCH WHEN SET INHIBITS RELOCATION OF THE PROGRAM ABOVE 28K.
SW8	LOAD PDP-11/45	THIS SWITCH WHEN SET LOADS THE MICRO
SW7-0	MICRO BREAK-	BREAK REGISTER WITH THE VALUE SET INTO

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
SWITCH SETTINGS

	REGISTER	SW7-0 AT THE BEGINNING OF EACH SUBTEST.
SW7	INHIBIT END OF PASS TYPEOUT	THIS SWITCH WHEN RESET INHIBITS THE END OF PASS TYPEOUT (THE QUICK BROWN FOX...).
SW6	INHIBIT CLOCK INTERRUPTS	THIS SWITCH WHEN SET WILL TURN THE CLOCK(S) OFF.
SW05	ENABLE RELOCATION VIA ALL AVAIL. DISKS	THIS SWITCH WILL CAUSE PROGRAM RELOCATION VIA ALL AVAILABLE DISKS ROUND ROBIN STYLE, I.E., FIRST RELOCATION VIA CP, THEN RK, RF, RP, ETC.
SW04	ENABLE RANDOM DISK ADDRESS SELECTION FOR RELOCATION	IF NOT ENABLED ALL DISK RELOCATION TRANSFERS BEGIN AT DISK ADDRESS 0.
SW03	ENABLE RELOCATION VIA I/O DEVICE	
SW02-SW00	DEVICE CODES	THESE SWITCHES WHEN SET CAUSE THE PROGRAM TO RELOCATE THE TEST CODE USING THE DEVICE SPECIFIED BELOW:

VALUE	DEVICE
0	CP
1	RK
2	RF
3	RP
4	RC
5	RP04
6	RS04
7	RK06

NOTE

WHEN RELOCATING VIA AN I/O DEVICE, SET IN THE VALUE TO SELECT THE DEVICE THEN SET SWITCH 3.

3.1 11/45 DISPLAY REGISTER

THE PASS COUNT IS DISPLAYED IN BITS 00-02. THE SECTION NUMBER IS DISPLAYED IN BITS 06-03. THE MOST SIGNIFICANT BYTE OF THE BASE ADDRESS (CONTENTS OF FRSTAD) OF THE SECTION OF CODE BEING EXECUTED IS DISPLAYED IN BITS 15-08. WHEN MEMORY MANAGEMENT IS ENABLED THE CONTENTS OF KIPAR2 IS DISPLAYED. KIPAR2 CONTAINS THE BASE PAGE

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
SWITCH SETTINGS

ADDRESS OF THE CODE BEING EXECUTED.

NOTE

THE RF11 DATA BUFFER REGISTER ALSO DISPLAYS THE ABOVE INFORMATION IF THE RF IS SELECTED.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
ERRORS

CHAPTER 4

ERRORS

IF AN ERROR IS DETECTED, THE PROGRAM WILL TRAP TO THE ERROR HANDLING ROUTINE (ERROR). IF ERROR TYPEOUT IS ENABLED, THIS ROUTINE WILL TYPE THE PC AND THE PROCESSOR STATUS AT THE TIME OF THE ERROR. ALSO, (IF REQUIRED), THE ORIGINAL PC (WHERE THE PC WAS RELOCATED FROM).

4.1.1 ERROR PRINTOUT FORMAT (CP ERROR)

```
PASS # AAAA VPC=BBBBBB PSW=DDDDDD

                    OR

PASS # AAAA VPC=BBBBBB PSW=DDDDDD RPC=CCCCCC

                    OR

PASS # AAAA VPC=BBBBBB PSW=DDDDDD PPC=EEEEEE
```

WHERE:
VPC=VIRTUAL PC
RPC=PC OF ORIGINAL CODE
PPC=PHYSICAL PC
AAAA=PASS COUNT
BBBBBB=VIRTUAL PC AT THE TIME OF THE ERROR
CCCCCC=PC OF THE ORIGINAL CODE RELOCATED
DDDDD=PSW AT THE TIME OF THE ERROR
EEEEEE=PHYSICAL PC AT THE TIME OF THE ERROR.

THE FIRST ERROR FORMAT SHOWS AN ERROR DETECTED WHEN THE PROGRAM IS NOT RELOCATED, AND, IN THIS INSTANCE VPC=PPC. THE ERROR IS PROBABLY A CP ERROR.

THE SECOND ERROR FORMAT SHOW AN ERROR DETECTED WHEN THE PROGRAM IS RELOCATED BELOW 28K, AND, IN THIS INSTANCE VPC=PPC. THE ERROR IS PROBABLY DUE TO A MEMORY ERROR.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
ERRORS

THE THIRD ERROR FORMAT SHOWS AN ERROR DETECTED WHEN THE PROGRAM IS RELOCATED ABOVE 28K. THE ERROR IS PROBABLY DUE TO A MEMORY ERROR. NOTE THAT VPC IS THE PC OF THE ORIGINAL CODE.

TO OBTAIN THE 'PHYSICAL' PC (11/45 ONLY), SET THE ADDRESS SELECTOR TO THE KLI POSITION. LOAD ADDRESS AND EXAMINE THE PC ADDRESS, SET THE ADDRESS SELECTOR TO 'PROGRAM PHYSICAL'. THE ADDRESS DISPLAYED IS THE PHYSICAL PC. ON THE 11/40 TO OBTAIN THE 'PHYSICAL' PC ADD THE CONTENTS OF KIPAR2 OR KIPAR3 TO THE VIRTUAL PC.

NOTE

USE CAUTION WHEN EXAMINING/DEPOSITING INTO ADDRESSES WHEN MEMORY MANAGEMENT IS ENABLED.

4.1.2 ERROR PRINTOUT FORMAT (DEVICE ERROR)

PASS # AAAA VPC=BBBBBB XX ERROR

111111 222222 333333 444444 555555 666666

WHERE:
VPC=VIRTUAL PC
AAAA=PASS COUNT
BBBBBB=VIRTUAL PC AT TIME OF ERROR
XX=TWO LETTER DEVICE IDENTIFIER
111111-666666=CONTENTS OF DEVICE REGISTER

4.1.3 ERROR PRINTOUT FORMAT (PARITY ERROR)

PARITY ERROR

THE PC AT THE TIME OF THE ERROR IS TYPED AS SHOWN IN SECTION 4.1.1.

MEMORY ADDRESS = XXXXXX, GOOD DATA = XXXXXX, BAD DATA = XXXXXX.

NOTE

THE ADDRESS TYPED IS THE 18 BIT PHYSICAL ADDRESS.

4.1.4 ERROR PRINTOUT FORMAT (RELOCATION ERROR)

PASS # AAAA VPC=BBBBBB MM ERROR

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
ERRORS
FROM ADRS=XXXXXX DATA=XXXXXX TO ADRS=XXXXXX DATA=XXXXXX

NOTE

THE ADDRESSES ARE 18 BIT PHYSICAL ADDRESSES "FROM" ADDRESS IS IN R0 "TO" ADDRESS IS IN R2.

4.2 PARITY ERROR DETECTION

IF A PARITY ERROR IS DETECTED THE PROGRAM WILL TYPE A MESSAGE "PARITY ERROR". PRINT THE PC AT THE TIME OF THE ERROR (VIA HLT) AND SCAN MEMORY FOR THE PARITY ERROR. WHEN THE PROGRAM FINDS THE PARITY ERROR IT WILL TYPE A MESSAGE "MEMORY ADDRESS IS BBBBBB". WHEN THE ADDRESS IS FOUND THE FAILING ADDRESS IS SCANNED WITH A BINARY COUNT PATTERN. WHEN THE PROGRAM FINDS THE FAILING DATA THE GOOD DATA AND BAD DATA ARE TYPED. IF THE PROGRAM DOES NOT FIND THE PARITY ERROR ON THE ADDRESS/DATA SCAN IT WILL TYPE A MESSAGE "PARITY ERROR NOT DETECTED ON ADDRESS/DATA SCAN". THE PROGRAM IS THEN RESTARTED.

4.3 ERROR LOOPING

THE SUBTEST DETECTING THE ERROR MAY BE LOOPED INDEFINITELY BY SETTING SW14. SETTING SW13 WILL INHIBIT THE TYPEOUT AND ALLOW SCOPING THE FAULTY SIGNAL(S).

4.4 UNPREDICTED ERRORS

THE PROGRAM MAY ON OCCASSION DETECT A MEMORY ERROR THE RESULTS OF WHICH WERE NOT PREDICTABLE IN WHICH CASE THE PROGRAM MAY BEHAVE UNPREDICTABLY. WHEN THIS HAPPENS THE USER MUST RETRACE THE PROGRAM STEPS TO RESOLVE WHERE THE ERROR OCCURRED. THE FOLLOWING ITEMS SHOULD BE CONSIDERED AND MAY BE OF USE WHEN RETRACING A FAILURE OF THIS NATURE.

1. HALT THE PROGRAM (IF NECESSARY).

2. EXAMINE RELR1
   ADDRESS RELR1 (1006) CONTAINS THE UNRELOCATED VALUE OF THE PC OF THE LAST TEST THAT WAS SUCCESSFULLY EXECUTED.

3. EXAMINE FACTOR
   ADDRESS FACTOR (1004) CONTAINS THE RELOCATION FACTOR.

4. EXAMINE ALL LOCATIONS STARTING WITH THE ADDRESS SPECIFIED IN R1/R11 (IF PSW BIT11 = 0/1) COMPARING THEIR CONTENTS WITH THE CONTENTS OF THE CORRESPONDING UNRELOCATED CODE (SPECIFIED IN

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
ERRORS

1006) AS SHOWN IN THE LISTING. EXAMINE AND COMPARE UNTIL EITHER A DIFFERENCE IN INSTRUCTION (I.E., THE ERROR) OR THE NEXT 'SCOPE' IS SEEN.

1A. EXAMINE THE STACK (R6)
THE TOP WORD ON THE STACK CONTAINS THE PC AT THE TIME OF THE TRAP. IF THE PC IS GREATER THAN THE LAST LOCATION IN THE LISTING THEN -

2A. EXAMINE LOCATION 1004 (FACTOR)
THIS LOCATION CONTAINS THE PROGRAM RELOCATION FACTOR WHICH, WHEN SUBTRACTED FROM THE PC GIVES THE PC OF THE ORIGINAL CODE.

4.5 TRAP TO LOCATION 4

IF A TRAP TO LOCATION 4 OCCURS THE PROGRAM WILL TYPE: "TRAP TO 4". THEN THE ERROR PRINTOUT INFORMATION (AS IN 4.1.1) WILL BE TYPED.

NOTE

THE PC TYPED WILL BE THE PC-2 AT THE TIME THAT THE TRAP OCCURED. THE PROGRAM WILL THEN RESTART AT THE LAST 'SCOPE'

4.6 TRAP TO LOCATION 10

IF A TRAP TO LOCATION 10 (RESERVED INSTRUCTION) OCCURS THE PROGRAM WILL TYPE: "RESERVED INSTRUCTION TRAP" AND THE ADDITIONALSIK IINFORMATION INSTRUCTION (AS IN 4.1.1). THE PC TYPED WILL BE THE PC-2 AT THE TIME OF THE TRAP. THE PROGRAM WILL RESTART AT THE LAST 'SCOPE'

4.7 MEMORY MANAGEMENT (KT11) ABORT

IF A KT11 ABORT (TRAP AT 250) OCCURS, THE PROGRAM WILL TYPE A MESSAGE "KT11 ABORT". THEN THE ERROR PRINTOUT INFORMATION (AS IN 4.1.1) WILL BE TYPED.

NOTE

THE PC TYPED WILL BE THE CONTENTS OF SR2 AT THE TIME THAT THE TRAP OCCURRED. THE PROGRAM WILL THEN RESTART A THE LAST SCOPE

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
ERRORS

4.8 ERROR DISCUSSION

AN ERROR DETECTED WHEN THE PROGRAM IS NOT RELOCATED IS LIKELY TO BE A CP MALFUNCTION. AN ERROR DETECTED WHEN THE PROGRAM IS RELOCATED BETWEEN 40000 AND 160000 COULD BE EITHER A CP OR MEMORY MALFUNCTION. AN ERROR DETECTED WHEN THE PROGRAM IS RELOCATED ABOVE 160000 (28K) IS MOST LIKELY A MEMORY MALFUNCTION. THE MEMORY EXERCISER (DZQMB-) SHOULD BE RUN IF A MEMORY FAILURE IS SUSPECTED, SELECTING ONLY THOSE BANK(S) DEEMED BAD.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
SUBROUTINE ABSTRACTS

CHAPTER 5

SUBROUTINE ABSTRACTS

5.1 SCOPEA

THE SCOPEA ROUTINE IS ENTERED BY THE SCOPE (EMT) INSTRUCTION AND IS EXECUTED AT THE START OF EACH SUBTEST. THE ROUTINE MONITORS SW14, SW11 AND SW8 AND TAKES APPROPRIATE ACTION. ALSO, THIS ROUTINE STORES IN R1/R11 THE FIRST ADDRESS OF THE SUBTEST BEING ENTERED.

5.2 .HLT

THE .HLT ROUTINE IS ENTERED BY THE HLT (TRAP) INSTRUCTION, AND IS EXECUTED WHEN A PREDICTABLE ERROR IS DETECTED. THIS ROUTINE MONITORS SW15, SW13, AND SW10.

5.3 PROGRAM RELOCATION

FOUR ROUTINES ARE USED TO PERFORM PROGRAM RELOCATION. THE GENERAL FLOW IS AS FOLLOWS:

IF BELOW 28K	THE RELOC ROUTINE IS CALLED AFTER A SECTION OF CODE HAS BEEN EXECUTED. IF AN I/O DEVICE IS SELECTED, SUBROUTINE IODEV IS CALLED AND THE ROUTINE WAITIO IS EXECUTED WHILE THE DEVICE IS TRANSFERRING CODE.
IF ABOVE 28K	THE STMM ROUTINE IS CALLED AFTER THE ENTIRE PROGRAM HAS BEEN EXECUTED. IF AN I/O DEVICE IS SELECTED, SUBROUTINE IODEV IS CALLED AND THE ROUTINE WAITIO IS EXECUTED WHILE THE DEVICE IS TRANSFERRING CODE.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
SUBROUTINE ABSTRACTS

5.3.1 RELOC

THE RELOC ROUTINE IS ENTERED BY A MOV #RELOC,PC INSTRUCTION. THIS ROUTINE RELOCATES THE PROGRAM CODE THROUGHOUT MEMORY AND 'JUMPS' TO THE RELOCATED CODE AFTER IT HAS BEEN MOVED SUCCESSFULLY. THE CODE IS RELOCATED BY 'MOVING' THE CODE VIA MOV INSTRUCTIONS. IF AN I/O DEVICE IS SELECTED VIA SWITCH REGISTER <3-0>, THE CODE IS RELOCATED BY WRITING THE CODE ONTO THE I/O DEVICE AND READING THE CODE BACK INTO ITS RELOCATED POSITION. IF THE CODE CANNOT BE RELOCATED (BECAUSE OF INSUFFICIENT MEMORY) THE ROUTINE 'JUMPS' TO THE NEXT SECTION OF UNRELOCATED PROGRAM CODE. THE CODE MOVED IS LESS THAN 1K ((4000) BYTES). AT THE START AND END OF EACH SECTION OF CODE TO BE MOVED IS A SECTION OF CODE WHICH ESTABLISHES THE FIRST ADDRESS OF THE CODE TO BE MOVED, AND SETS A SCOPE POINTER (R1/R11) AND ALSO A SECTION WHICH ESTABLISHES THE LAST ADDRESS AND 'JUMPS' TO THE RELOCATION (RELOC) ROUTINE. EACH SECTION OF CODE IS IDENTIFIED AS SHOWN BELOW.

```
;0000000000FIRST ADDRESS TO BE RELOCATED0000000000

        CODE TO BE MOVED AND EXECUTED

;0000000000LAST ADDRESS OF CODE TO BE RELOCATED0000000000
```

THE RELOC ROUTINE DOES NOT RELOCATE PROGRAM CODE INTO THE LAST 1000(OCTAL) BYTES OF MEMORY, THUS PRESERVING THE LOADERS. THIS ROUTINE MONITORS SW12, SW05, AND SW03.

5.3.2 RELOCATION ABOVE 28K (STMM)

THE STMM SUBROUTINE RELOCATES THE PROGRAM CODE ABOVE 28K IF MEMORY AND THE KT OPTION ARE AVAILABLE. THE ROUTINE MOVES THE CODE AT 0-8K UPWARDS TO ADDRESSES ABOVE 28K. EACH SUCCEEDING RELOCATION IS TO MEMORY 1K GREATER THAN THE LAST. THE PROGRAM IS EXECUTED IN ALL CASES FROM VIRTUAL MEMORY ADDRESSES 0-37776, HOWEVER, THE PHYSICAL ADDRESS CHANGES BY 1K (4000) ON EACH RELOCATION.

NOTE

THE 'VIRTUAL' LIGHT (11/40) WILL BE ON
WHEN THE PROGRAM IS EXECUTING ABOVE 28K.

THIS ROUTINE MONITORS SW12, SW09, SW05, AND SW03.

5.3.3 IODEV

THE IODEV SUBROUTINE IS CALLED FROM EITHER THE RELOC OR STMM ROUTINES WHENEVER AN I/O DEVICE IS SELECTED TO PERFORM PROGRAM RELOCATION. THIS ROUTINE OBTAINS THE PHYSICAL BUS ADDRESS FOR READ AND WRITE AND

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
SUBROUTINE ABSTRACTS

THE BYTE COUNT FROM THE CALLING ROUTINE. THE DEVICE TO BE USED IS OBTAINED FROM LOCATION DEV. THE CODE TO BE RELOCATED IS WRITTEN FROM ITS PRESENT POSITION AND THEN READ INTO THE RELOCATED POSITION. IF A DEVICE ERROR OCCURS THE ERROR IS REPORTED AND THE OPERATION IS RETRIED UP TO THREE TIMES.

5.3.4 WAITIO

THE PURPOSE OF THE WAITIO ROUTINE IS TO REFERENCE VIA THE CP THE SAME MEMORY LOCATIONS AS THE DEVICE DURING THE NPR TRANSFERS.

5.3.5 DSKADR

THE DSKADR SUBROUTINE IS CALLED FROM THE IODEV ROUTINE. IT GENERATES A RANDOM DISK ADDRESS FOR THE SELECTED DISK IF SW04 IS SET. OTHERWISE IT GENERATES A '0' DISK ADDRESS. THE GENERATED RANDOM ADDRESSES ARE LIMITED (SO DISK OVERFLOW WILL NOT OCCUR) BY THE TABLE ADRTAB.

5.4 CLOCK INTERRUPT

THE CLOCK INTERRUPT FOR THE LINE AND PROGRAMMABLE CLOCKS INCREMENT LOCATIONS LTICKS AND PTICKS ON EACH INTERRUPT. THIS ROUTINE MONITORS SW06.

5.5 END

THIS ROUTINE IS ENTERED AT THE COMPLETION OF EACH PASS IT SETS UP (LOADS NEW PROCESSOR STATUS) FOR THE NEXT PASS AND PRINTS AN END OF PASS MESSAGE:

```
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK
0123456789 PASS # AAAA
```

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
MISCELLANEOUS

CHAPTER 6

MISCELLANEOUS

6.1 EXECUTION TIME

THE EXECUTION TIME IS HIGHLY VARIABLE (DEPENDENT ON PROCESSOR, TYPE OF MEMORY, AND AMOUNT OF MEMORY). HOWEVER, WHEN THE PROGRAM IS RUNNING SUCCESSFULLY THERE IS A NOTICEABLE 'FLICKER' DISPLAYED ON THE CONSOLE LIGHT PATTERN. THE 'FLICKER' WILL DIM WHEN 'T' BIT TRAP PASSES (EVERY ODD PASS) ARE RUNNING, THE PROGRAM SHOULD BE RUN FOR A MINIMUM OF:

4 PASSES (PASS # 0003) 11/40
8 PASSES (PASS # 0007) 11/45

SOME TYPICAL TIMES FOLLOW:

PFP-11/45 WITH 104K MEMORY (96K CORE, 8K MOS)-24 MINS
PDP-11/45 WITH 48K MEMORY-10 MINS

6.2 PASS MODIFICATION

THE PSW OF THE PASS MAY BE MODIFIED BY PATCHINC INTO LOCATION PSWTAB+2 THE DESIRED PSW. FOR EXAMPLE PATCHING 040000 INTO PSWTAB+2 CAUSES THE PROGRAM TO RUN IN SUPERVISOR MODE ON THE SECOND PASS.

6.3 I/O DEVICE ADDRESS MODIFICATION

TO MODIFY THE PROGRAM ADDRESS OF THE I/O DEVICES ON THE UNIBUS PATCH THE APPROPRIATE DEVICE TABLE (SEE LISTING TABLE OF CONTENTS - DEVICE TABLES) AND ALSO THE APPROPRIATE TABLE ENTRY AT 'REGADR' IN THE ERROR SERVICE ROUTINE.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
MISCELLANEOUS

6.4 MEMORY MODIFICATION

THE PROGRAM MAY BE MODIFIED TO PROVIDE EXTENDED MEMORY EXERCISING. ESSENTIALLY THE MODIFICATION INCREASES THE TEST ITERATION COUNT WHICH CAUSES TEST CODE TO BE EXECUTED IN MEMORY FOR A LONGER PERIOD OF TIME. NOTE THAT THIS MODIFICATION WILL INCREASE THE RUN TIME SUBSTANTIALLY. THE MODIFICATION IS:

PATCH	LOCATION	FROM	TO
	5454	020040	100200

6.5 USER DEFINED RELOCATION LIMITS

THE PROGRAM WILL REQUEST A LOWER AND UPPER LIMIT FOR RELOCATION. THE LIMITS MUST BE BETWEEN THE LAST LOCATION IN THE LISTING AND 157776. THE PROGRAM WILL EXECUTE IN THE LOWER 4K (0-17776) AND THE LIMITS SPECIFIED. THE STARTING ADDRESS IS 204. TO RETAIN PREVIOUSLY SPECIFIED LIMITS, START AT 210.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
PROGRAM DESCRIPTION

CHAPTER 7

PROGRAM DESCRIPTION

THE PROGRAM IS DIVIDED INTO FOUR SECTIONS OF POSITION INDEPENDENT RELOCATABLE TEST CODE. EACH SECTION IS APPROXIMATELY 1K WORDS LONG. (EXCEPT SECTION 0).

WHEN THE PROGRAM IS INITIALLY LOADED STARTED IT WILL IDENTIFY ITSELF AND TYPE THE CP AND CP OPTIONS AVAILABLE INDICATOR WORD (OPT.CP). THE CONTENTS OF OPT.CP CONTAIN THE FOLLOWING INDICATORS:

BIT15 = 1/0 =	MEMORY MANAGEMENT OPTION AVAILABLE/NOT AVAILABLE
BIT14 = 1/0 =	EIS AVAILABLE/NOT AVAILABLE

NOTE

EIS IS ALWAYS AVAILABLE ON PDP-11/45.

BIT13 = 1/0 =	11/45 FPP AVAILABLE/NOT AVAILABLE
BIT12 = 1/0 =	11/40 FIS AVAILABLE/NOT AVAILABLE
BIT11 = 1/0 =	STACK LIMIT (11/40 KT OPTION) AVAILALE/NOT AVAILABLE
BIT10 = 1/0 =	KW11-P AVAILABLE/NOT AVAILABLE
BIT09 = 1/0 =	KW11-L AVAILABLE/NOT AVAILABLE
BIT08 = 1/0 =	CONSOLE TTY AVAILABLE/NOT AVAILABLE
BITS 07-00 =	06 = 11/45, 04 = 11/40.

PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
PROGRAM DESCRIPTION

SECTION 0 — THIS SECTION CAUSES A 256 WORD 3X OR 9 WORST CASE NOISE TEST PATTERN TO BE RELOCATED THROUGHOUT MEMORY 0 - 28K.

NOTE

THIS SHOULD NOT BE CONSTRUCTED TO BE A MEMORY TEST.

SECTION 1 — THIS SECTION TESTS THE UNARY INSTRUCTION SET EXECUTING EACH UNARY INSTRUCTION IN EACH ADDRESS MODE (EXCLUDING UNARY INSTRUCTIONS USING ADDRESS MODE 7).

SECTION 2 — THIS SECTION TESTS THE UNARY INSTRUCTIONS USING ADDRESS MODE 7 AND BINARIES IN ALL ADDRESS MODES (EXCLUDING BINARY BYTES OPS USING ADDRESS MODE 7).

SECTION 3 — THIS SECTION TESTS BINARY BYTE OPS USING ADDRESS MODE 7, JMP, JSR AND PROGRAM TRAP (IOT, TRAP, AND EMT) INSTRUCTIONS.

SECTION 4 — THIS SECTION CHECKS THAT EACH BIT IN THE PROCESSOR STATUS WORD (PSW) CAN BE SET CLEARED, RESERVED INSTRUCTION, AND ODD ADDRESS TRAPS.

SECTION 5 — THIS SECTION CHECKS THE SXT, XOR, SOB, MARK, RTT AND RTT INSTRUCTIONS.

SECTION 6 — THIS SECTION CHECKS THE ASH, ASHC, MUL, DIV, SPL INSTRUCTIONS AND THE PROGRAM INTERRUPT REQUEST (PIRQ) LOGIC.

SECTION 7 — THIS SECTION CHECKS THE STACK LIMIT REGISTER (KJ-11 OPTION ON 11/40), AND MEMORY MANAGEMENT ABORT LOGIC (IF SYSTEM HAS MORE THAN 32K OF MEMORY).

FOLLOWING SECTION 7 ARE TWO ROUTINES TO CHECK THE TELETYPE PRINTER LOGIC AND A ROUTINE TO START EITHER THE KW11-P OR THE KW11-L CLOCK. IF EITHER THE KW11-P OR THE KW11-L IS AVAILABLE THE PRIORITY ARBITRATION LOGIC IS TESTED.

THE PROGRAM THEN RELOCATES TO 160000 (IF AVAILABLE) AND RESTARTS. THE PROGRAM CONTINUES RELOCATING BY INCREMENTS OF 4000 BYTES (1K) UNTIL THE END OF MEMORY IS REACHED. RELOCATION OF THE PROGRAM THROUGHOUT ALL MEMORY CONSTITUTES A PASS. WHEN THE PROGRAM IS EXECUTING ABOVE 28K, YOU WILL HEAR SEVERAL 'KERCHUNKS' ON THE TELETYPE. THE 'KERCHUNKS' ARE CAUSED BY THE TELETYPE TEST FOLLOWING SECTION 7 MENTIONED ABOVE.

```
                        PDP-11/40 AND PDP-11/45 INSTRUCTION EXERCISER
                        PROGRAM DESCRIPTION


                        UPON COMPLETION OF A PASS THE PROGRAM RESTARTS USING A  NEW  PROCESSOR
                        STATUS DEPENDING ON THE TYPE OF PROCESSOR AND THE PASS COUNT.



                        7.1  STACK POINTER

                        THE STACK POINTER IS SET AT 500.

                                                        NOTE

                                                IF THE PROGRAM IS RUNNING IN EITHER USER
                                                OR  SUPERVISOR MODE, THE USER/SUPERVISOR
                                                STACK POINTER IS  SET  TO  500  AND  THE
                                                KERNEL  STACK POINTER IS SET TO 600. THE
                                                KERNEL STACK POINTER IS  USED  ONLY  FOR
                                                THE    SCOPE    HIT,    TTY,    AND    CLOCK
                                                TRAP/INTERRUPT ROUTINES.



                        7.2  POWER FAILURE

                        A POWER FAIL SERVICE ROUTINE IS INCORPORATED IN THE TEST.  WHEN  USING
                        THIS  PROGRAM THE POWER SHOULD BE TURNED OFF WHEN RUNNING TO CHECK THE
                        POWER FAIL LOGIC.  WHEN THE POWER FAILS THE PROGRAM WILL TYPE:

                                  POWER FAILED

                        AND RESTART THE PROGRAM AT THE BEGINNING (START).
                           !
                                .NLIST  MD,MC,TOC
                                .LIST   ME
                                .ABS
                                .TITLE  DCQKCG  11/40-11/45 CPU EXERCISER
                                .SBTTL  SWITCH SETTING
                        ;       SW15---HALT ON ERROR
                        ;       SW14---LOOP TEST
                        ;       SW13---INHIBIT ERROR TYPEOUT
                        ;       SW12---SEE NOTE BELOW
                        ;       SW11---INHIBIT TEST ITERATIONS
                        ;       SW10---RING BELL ON ERROR
                        ;       SW09---SEE NOTE BELOW
                        ;       SW08---LOAD MICRO BRAK REGISTER WITH SW07-SW00
                        ;       SWC7---TYPE END OF PASS MESSAGE
                        ;       SW06---DISABLE CLOCKS
                        ;       SW05---RELOCATE USING ALL DEVICES ROUND ROBIN STYLE
                        ;       SW04---USE RANDOM DISK ADDRESS FOR RELOCATION
                        ;       SW03---RELOCATE USING DEVICE SELECTED IN SW02-SW00
                        ;       SW02-SW00---    0=CP
                        ;                       1=RK
```

```
1041                                    ;                       2=RF
1042                                    ;                       3=RP
1043                                    ;                       4=RC
1044                                    ;                       5=RP04/05/06
1045                                    ;                       6=RS03/RS04
1046                                    ;                       7=RK06
1047                                    ;NOTE BELOW: SW12 AND SW09 CONTROL PROGRAM RELOCATION DESRIBED BELOW:
1048                                    ;       SW12    SW09    RELOCATION
1049                                    ;       1       0       NONE
1050                                    ;       0       1       NO RELOCATION ABOVE 28K
1051                                    ;       1       1       NOT USED (DO NOT USE)
1052                                    ;       0       0       ALL MEMORY
1053
1054
1055
1056                                            .SBTTL  DEFINITIONS & ASSIGNMENTS
1057                                    ;GENERAL REGISTER ASSIGNMENTS
1058            000000                          R0=%0
1059            000001                          R1=%1
1060            000002                          R2=%2
1061            000003                          R3=%3
1062            000004                          R4=%4
1063            000005                          R5=%5
1064            000006                          SP=%6
1065            000007                          PC=%7
1066            000000                          R10=%0
1067            000001                          R11=%1
1068            000002                          R12=%2
1069            000003                          R13=%3
1070            000004                          R14=%4
1071            000005                          R15=%5
1072
1073                                    ;FLOATING POINT REGISTERS
1074            000000                          AC0=%0
1075            000001                          AC1=%1
1076            000002                          AC2=%2
1077            000003                          AC3=%3
1078            000004                          AC4=%4
1079            000005                          AC5=%5
1080
1081                                    ;STACK POINTER REGISTERS
1082            000006                          KSP=%6                  ;KERNEL STACK POINTER
1083            000006                          SSP=%6                  ;SUPERVISOR STACK POINTER
1084            000006                          USP=%6                  ;USER STACK POINTER
1085
1086                                    ;STATUS REGISTER (PSW) BIT ASSIGNMENTS
1087            000001                          C=1                     ;C BIT
1088            000002                          V=2                     ;V BIT
1089            000004                          Z=4                     ;Z BIT
1090            000010                          N=10                    ;N BIT
1091            000020                          T=20                    ;'T' BIT
1092            000340                          PRTY7=340               ;PRIORITY LEVEL 7
1093            000300                          PRTY6=300               ;PRIORITY LEVEL 6
1094            000240                          PRTY5=240               ;PRIORITY LEVEL 5
1095            000200                          PRTY4=200               ;PRIORITY LEVEL 4
1096            000140                          PRTY3=140
```

```
1097    000100                  PRTY2=100               ;PRIORITY LEVEL 2
1098    000000                  KM=000000               ;KERNEL MODE
1099    040000                  SM=040000               ;SUPERVISORY MODE
1100    140000                  UM=140000               ;USER MODE
1101    000000                  PKM=000000              ;PREVIOUS KERNEL MODE
1102    010000                  PSM=010000              ;PREVIOUS SUPERVISORY MODE
1103    030000                  PUM=030000              ;PREVIOUS USER MODE
1104    004000                  REG=004000              ;SELECT R10-R15
1105
1106                    ;VECTOR ADDRESSES
1107    000004                  ERRVEC= 4               ;ADDRESS OF ERROR VECTOR
1108    000010                  RESVEC= 10              ;ADDRESS OF RESERVED INST. TRAP VECTOR
1109    000014                  TBITVEC=14              ;ADDRESS OF 'T' BIT TRAP VECTOR
1110    000014                  TRTVEC= 14              ;ADDRESS OF 'TRACE' TRAP VECTOR
1111    000014                  BPTVEC= 14              ;ADDRESS OF 'BREAKPOINT' TRAP VECTOR
1112    000020                  IOTVEC= 20              ;ADDRESS OF IOT TRAP VECTOR
1113    000024                  PFVEC=  24              ;ADDRESS OF POWER FAIL TRAP VECTOR
1114    000030                  EMTVEC= 30              ;ADDRESS OF EMT VECTOR
1115    000034                  TRAPVEC=34              ;ADDRESS OF TRAP VECTOR
1116    000060                  TKVEC=  60              ;ADDRESS OF KEYBOARD INTERRUPT VECTOR
1117    000064                  TPVEC=  64              ;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
1118    000070                  PRVEC=  70              ;HIGH SPEED READER INTERRUPT VECTOR
1119    000074                  PPVEC=  74              ;HIGH SPEED PUNCH INTERRUPT VECTOR
1120    000100                  LKVEC=  100             ;ADDRESS KW11-L LINE CLOCK INT. VECTOR
1121    000104                  PLKVEC= 104             ;ADDRESS OF KW11-P CLOCK INT VECTOR
1122    000204                  RFVEC=  204             ;RF OR RS04 VECTOR
1123    000204                  RSVEC=  204
1124    000210                  RCVEC=  210             ;RC VECTOR
1125    000220                  RKVEC=  220             ;RK DISK VECTOR
1126    000210                  RK6VEC= 210             ;RK06 VECTOR
1127    000240                  PIRVEC= 240             ;ADDRESS OF PIRQ VECTOR
1128    000244                  FPEVEC= 244             ;ADDRESS OF FLOATING POINT INT. VECTOR
1129    000250                  MMVEC=  250             ;ADDRESS OF MEM MGMT ERROR TRAP VECTOR
1130    000254                  RPVEC=  254             ;RP VECTOR
1131    000254                  RP4VEC= 254             ;RP04 VECTOR
1132
1133                    ;REGISTER ADDRESSES
1134    177776          PSW=    177776          ;ADDRESS OF STATUS REGISTER
1135    177774          SLR=    177774          ;ADDRESS OF STACK LIMIT REGISTER
1136    177772          PIRQ=   177772          ;ADDRESS OF PROGRAM INTERRUPT REQUEST
1137    177770          UBREAK= 177770          ;ADDRESS OF MICRO BREAK REGISTER
1138    177766          CPUERR= 177766
1139    177744          ERRREG= 177744
1140    177546          LKS=    177546          ;ADDRESS OF KW11-L STATUS REG.
1141    177550          PRS=    177550          ;ADDRESS OF HIGH SPEED READER CSR
1142    177552          PRB=    177552          ;ADDRESS OF HIGH SPEED READER DATA BUF
1143    177554          PPS=    177554          ;ADDRESS OF HIGH SPEED PUNCH CSR
1144    177556          PPB=    177556          ;ADDRESS OF HIGH SPEED PUNCH BUFFER
1145    177560          TKS=    177560          ;ADDRESS OF KEYBOARD CSR
1146    177562          TKB=    177562          ;ADDRESS OF KEYBOARD BUFFER
1147    177564          TPS=    177564          ;ADDRESS OF TELEPRINTER CSR
1148    177566          TPB=    177566          ;ADDRESS OF TELEPRINTER BUFFER
1149    177572          SR0=    177572          ;ADDRESS OF MEM MGMT REGISTER SR0
1150    177574          SR1=    177574          ;ADDRESS OF MEM MGMT REG SR1
1151    177576          SR2=    177576          ;ADDRESS OF MEM MGMT REGISTER SR2
1152    172516          SR3=    172516          ;ADDRESS OF MEM MGMT REGISTER SR3
```

```
1153    177570                  SWR=    177570          ;ADDRESS OF CONSOL SWITCH REGISTER
1154    177570                  DISPLAY=177570          ;ADDRESS OF CONSOL DISPLAY REGISTER
1155    177514                  LPS=    177514          ;ADDRESS OF LINE PRINTER STATUS REG
1156    177516                  LPB=    177516          ;ADDRESS OF LINE PRINTER DATA DUFFER
1157
1158                    ;RK REGISTERS
1159    177400                  RKDS=   177400          ;ADDRESS OF RK-11 DISK DRIVE STATUS REGISTER
1160    177402                  RKER=   177402          ;ADDRESS OF RK-11 DISK ERROR REGISTER
1161    177404                  RKCS=   177404          ;ADDRESS OF RK-11 DISK CONT. AND STATUS REG.
1162    177406                  RKWC=   177406          ;ADDRESS OF RK-11 DISK WORD COUNT REG.
1163    177410                  RKBA=   177410          ;ADDRESS OF RK-11 DISK BUS ADDRESS REG.
1164    177412                  RKDA=   177412          ;ADDRESS OF RK-11 DISK ADDRESS REG.
1165
1166                    ;RF REGISTERS
1167    177460                  RFDCS=  177460          ;ADDRESS OF RF-11 DISK CONT. AND STATUS REG.
1168    177462                  RFWC=   177462          ;ADDRESS OF RF-11 DISK WORD COUNT REG.
1169    177464                  RFCMA=  177464          ;ADDRESS OF RF-11 DISK MEMORY ADR.REG.
1170    177466                  RFDAR=  177466          ;ADDRESS OF RF-11 DISK ADDRESS REG.
1171    177470                  RFDAE=  177470          ;ADDRESS OF RF DAE REGISTER
1172
1173                    ;RC REGISTERS
1174    177440                  RCLA=   177440          ;ADDRESS OF RC-11 LOOK AHEAD REGISTER
1175    177442                  RCDA=   177442          ;ADDRESS OF RC-11 DISK ADDRESS REG.
1176    177446                  RCCS=   177446          ;ADDRESS OF RC-11 DISK CONT. AND STATUS REG.
1177    177450                  RCWC=   177450          ;ADDRESS OF RC-11 DISK WORD COUNT REG.
1178    177452                  RCCA=   177452          ;ADDRESS OF RC-11 CURRENT DISK ADR REG.
1179
1180                    ;RP04 REGISTERS
1181    176700                  RP4CS1= 176700          ;RP04 CS1 REGISTER
1182    176702                  RP4WC=  176702          ;WORD COUNT REGISTER
1183    176704                  RP4BA=  176704          ;BUS ADDRESS REGISTER
1184    176706                  RP4DST= 176706          ;DESIRED SECTOR/TRACK REGISTER
1185    176710                  RP4CS2= 176710          ;RP04 CS2 REGISTER
1186    176712                  RP4DS1= 176712          ;DRIVE STARTUS REGISTER #1
1187    176714                  RP4ER1= 176714          ;ERROR REGISTER #1
1188    176716                  RP4AS=  176716          ;ATTENTION SUMMARY
1189    176720                  RP4LA=  176720          ;LOOK AHEAD REGISTER
1190    176732                  RP4OF=  176732          ;OFFSET REGISTER
1191    176734                  RP4CA=  176734          ;DISK ADDRESS REGISTER
1192
1193                    ;RH11 MASS BUS CONTROLLER REGISTERS
1194    000000                  RHCS2=  0                       ;NOT DEFINED
1195
1196                    ;RP11C REGISTERS
1197    176710                  RPDS=   176710          ;ADDRESS OF RP DRIVE STATUS REGISTER
1198    176712                  RPER=   176712          ;ADDRESS OF RP ERROR REGISTER
1199    176714                  RPCS=   176714          ;ADDRESS OF RP CONTROL STATUS REGISTER
1200    176716                  RPWC=   176716          ;ADDRESS OF RP WORD COUNT REGISTER
1201    176720                  RPBA=   176720          ;ADDRESS OF RP BUS ADDRESS REGISTER
1202    176722                  RPCA=   176722          ;ADDRESS OF RP CYLINDER ADDRESS REGISTER
1203    176724                  RPDA=   176724          ;ADDRESS CF RP DISK ADDRESS REGISTER
1204
1205                    ;KW11-P REGISTERS
1206    172540                  PLKCSR= 172540          ;ADDRESS OF KW11-P CLOCK CSR
1207    172542                  PLKCSB= 172542          ;ADDRESS OF KW11-P COUNT SET BUFFER
1208    172544                  PLKCTR= 172544          ;ADDRESS OF KW11-P COUNTER
```

```
1209
1210                                    ;RSO4 REGISTERS
1211            172040                          RSCS1=  172040          ;CONTROL STATUS REGISTER
1212            172042                          RSWC=   172042          ;WORD COUNT REGISTER
1213            172044                          RSBA=   172044          ;BUS ADDRESS REGISTER
1214            172046                          RSDA=   172046          ;DISK ADDRESS REGISTER
1215            172050                          RSCS2=  172050          ;CONTROL STATUS #2
1216            172052                          RSDS=   172052          ;DRIVE STATUS REGISTER
1217            172054                          RSER=   172054          ;ERROR REGISTER #1
1218            172056                          RSAS=   172056          ;ATTENTION SUMMARY REGISTER
1219            172060                          RSLA=   172060          ;LOOK AHEAD REGISTER
1220
1221                                    ;RK06 REGISTERS
1222            177440                          RK6CS1= 177440          ;CONTROL AND STATUS REGISTER 1
1223            177442                          RK6WC=  177442          ;WORD COUNT REGISTER
1224            177444                          RK6BA=  177444          ;BUS ADDRESS REGISTER
1225            177446                          RK6DA=  177446          ;DISK ADDRESS REGISTER
1226            177450                          RK6CS2= 177450          ;CONTROL AND STATUS REGISTER 2
1227            177452                          RK6DS=  177452          ;DRIVE STATUS REGISTER
1228            177454                          RK6ER=  177454          ;ERROR REGISTER #1
1229            177456                          RK6OF=  177456          ;OFFSET REGISTER
1230            177460                          RK6DC=  177460          ;DESIRED CYLINDER REGISTER
1231
1232                                    ;MEMORY MANAGEMENT REGISTER ADDRESSES
1233            172300                          KIPDR0= 172300
1234            172302                          KIPDR1= 172302
1235            172304                          KIPDR2= 172304
1236            172306                          KIPDR3= 172306
1237            172310                          KIPDR4= 172310
1238            172316                          KIPDR7= 172316
1239            172340                          KIPAR0= 172340
1240            172342                          KIPAR1= 172342
1241            172344                          KIPAR2= 172344
1242            172346                          KIPAR3= 172346
1243            172350                          KIPAR4= 172350
1244            172356                          KIPAR7= 172356
1245
1246            177600                          UIPDR0=177600
1247            177602                          UIPDR1=177602
1248            177610                          UIPDR4=177610
1249            177614                          UIPDR6=177614
1250            177616                          UIPDR7=177616
1251            177640                          UIPAR0=177640
1252            177642                          UIPAR1=177642
1253            177650                          UIPAR4=177650
1254            177654                          UIPAR6=177654
1255            177656                          UIPAR7=177656
1256
1257            172200                          SIPDR0=172200
1258            172202                          SIPDR1=172202
1259            172210                          SIPDR4=172210
1260            172214                          SIPDR6=172214
1261            172216                          SIPDR7=172216
1262            172240                          SIPAR0=172240
1263            172242                          SIPAR1=172242
1264            172250                          SIPAR4=172250
```

```
1265            172254                  SIPAR6=172254
1266            172256                  SIPAR7=172256
1267            172320                  KDPDR0=172320
1268            177620                  UDPDR0=177620
1269            172220                  SDPDR0=172220
1270            172360                  KDPAR0=172360
1271            177660                  UDPAR0=177660
1272            172260                  SDPAR0=172260
1273
1274                                ;INITIAL STACK POINTER SETTING
1275            000500                  STKPTR= 500             ;PROGRAM STACK PTR
1276            000600                  KPTR=   600             ;KERNEL STACK PTR (USED BY KERNEL WHEN
1277                                                            ;PROG IS RUNNING IN OTHER THAN KERNEL MODE
1278
1279                                ;MISCELLANEOUS BIT ASSIGNMENTS (USED IN OPT.CP)
1280            100000                  KTOPT=  100000                  ;BELOW BIT ASSIGNMENTS ARE USED
1281            040000                  EISOPT= 040000                  ;IN THE CPCHK ROUTINE
1282            020000                  FPOPT=  020000                  ;A BIT FOR EACH OPTION PRESENT
1283            010000                  FISOPT= 010000                  ;IS SET IN OPT.CP (ODD BYTE)
1284            004000                  KJOPT=  004000
1285            002000                  PLKOPT= 002000
1286            001000                  LKOPT=  001000
1287            000400                  TTOPT=  000400
1288
1289                                ;MISCELLANEOUS BIT ASSIGNMENTS (USED IN OPT.DEV)
1290            000001                  RKOPT=  000001
1291            000002                  RFOPT=  000002
1292            000004                  RPOPT=  000004
1293            000010                  RCOPT=  000010
1294            000020                  RP40PT= 000020
1295            000040                  RSOPT=  000040
1296            000100                  RK60PT= 000100
1297
1298                                ;BIT ASSIGNMENTS USED IN OPTIONS
1299            000001                  PROPT=  000001
1300            000002                  PPOPT=  000002
1301
1302                                ;SWITCH DEFINITIONS
1303            010000                  SW12=   010000
1304            001000                  SW09=   001000
1305            000100                  SW06=   000100
1306            000040                  SW05=   000040
1307            000020                  SW04=   000020
1308            000010                  SW03=   000010
1309
1310            100000                  BIT15=  100000
1311            040000                  BIT14=  40000
1312            020000                  BIT13=  20000
1313            000400                  BIT8=   400
1314            000100                  BIT6=   100
1315            010000                  PIR4=   10000           ;LEVEL 4 PROGRAM INT. RQST. (FOR PIRQ)
1316
1317                                ;INSTRUCTION EQUATES
1318            104400                  HLT=    TRAP            ;HLT IS A TRAP INST TO THE ERROR ROUTINE
1319            104000                  SCOPE=  EMT             ;SCOPE IS AN EMT TRAP
1320            000004                  TYPE=   IOT
```

1321

```
1322                    000020                          .=IOTVEC
1323  000020  002736                                    .WORD   .TYPE                   ;SET IOT VECTOR TO TYPE ROUTINE
1324  000022  000340                                    .WORD   PRTY7
1325  000024  000610                                    .WORD   PDWN                    ;SET POWER FAIL VECTOR
1326  000026  000340                                    .WORD   PRTY7
1327  000030  001014                                    .WORD   SCOPEA
1328  000032  000200                                    .WORD   PRTY4
1329  000034  003416                                    .WORD   .HLT
1330  000036  000340                                    .WORD   PRTY7
1331                    000046                          .=46
1332  000046  032616                                    .WORD   LOGICAL                 ;ACT11 HOOK
1333                    000052                          .=52
1334  000052  040000                                    .WORD   40000
1335                    000060                          .=TKVEC
1336  000060  003314                                    .WORD   TKISR                   ;SET KEYBOARD VECTOR TO TKISR
1337  000062  000200                                    .WORD   PRTY4                   ;PRIORITY LEVEL 4
1338
1339                                                    .SBTTL  ENABLE PARITY & POWER FAIL ROUTINES
1340                    000120                          .=120
1341                                            ;ROUTINE TO SET PARITY ACTION ON PARITY MEMORIES
1342                    172100                  PARCSR= 172100          ;ADDRESS OF FIRST POSSIBLE PARITY REG
1343                    000114                  PARVEC= 000114          ;ADDRESS OF PARITY INTERRUPT VECTOR
1344
1345  000120  012737  004622  000114  .MAMF:   MOV     #.PARSRV,@#PARVEC       ;LOAD VECTOR
1346  000126  012737  000340  000116          MOV     #340,@#PARVEC+2         ;AND PRIORITY LEVEL
1347  000134  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;DO RTI ON TIME OUT TRAP
1348  000142  012700  172100                  MOV     #PARCSR,R0              ;GET FIRST POSSIBLE ADDRESS
1349  000146  012702  000001                  MOV     #1,R2                   ;SET REGISTER COUNTER
1350
1351  000152  012720  000001          1$:     MOV     #1,(R0)+                ;SET ACTION ENABLE (IF AVAIL)
1352                                            ;ABOVE INSTRUCTION WILL SET ACTION ENABLE IF MA/MF PARITY OR SET
1353                                            ;ODD PARITY AND HALT ON PARITY ERROR IF MOS PARITY
1354  000156  006302                          ASL     R2                      ;CHECK IF 16. REGISTERS HAVE
1355  000160  103374                          BCC     1$                      ;BEEN ENABLED
1356  000162  000207                  2$:     RTS     PC                      ;RETURN
1357
1358                    000200                          .=200
1359  000200  012707  005666                  MOV     #START,PC       ;GO TO START OF TEST
1360  000204  012707  005776                  MOV     #START1,PC      ;GO GET LOWER/UPPER RELOCATION BOUNDARY
1361  000210  012707  006050                  MOV     #START3,PC      ;START WITH LAST TYPED BOUNDARY LIMITS
1362
1363                    000244                          .=244
1364  000244  000246                          .WORD   246             ;SET FIS TRAP TO RETURN DIRECT
1365  000246  000002                          .WORD   RTI
1366                    000252                          .=252                   ;MEM MGMT PRIORITY ADDRESS
1367  000252  000340                          .WORD   340             ;SET AT LEVEL 7
1368                    000610                          .=610
1369                                            ;POWER FAIL SUBROUTINE
1370  000610  005737  000766          PDWN:   TST     @#OPT.CP
1371  000614  100002                          BPL     1$
1372  000616  005037  177572                  CLR     @#SR0
1373  000622  012737  000632  000024  1$:     MOV     #PUP,@#PFVEC
1374  000630  000000                          HALT
1375
1376                                            ;POWER UP SUBROUTINE
1377  000632  012737  000610  000024  PUP:     MOV     #PDWN,@#PFVEC   ;RESET POWER FAIL TRAP VECTOR TO POWER
```

E03


```
                                                                        ;DOWN ROUTINE ABOVE
000640  012706  000600                      MOV     #KPTR,SP          ;SET STACK PTR
000644  005027                              CLR     (PC)+
000646  000000                      1$:     .WORD   0                 ;KILL TIME
000650  005267  177772              2$:     INC     1$
000654  001375                              BNE     2$
000656  000004  000666                      TYPE,PFAIL
000662  000137  005666                      JMP     @#START           ;RESTART TEST

000666  005015  047520  042527  PFAIL:  .ASCIZ  <15><12>'POWER FAILED'<15><12>
000674  020122  040506  046111
000702  042105  005015      000
000707      015  050012  051101  PARERR: .ASCIZ  <15><12>'PARITY ERROR'<15><12>
000714  052111  020131  051105
000722  047522  006522  000012
```

```
                                        .SBTTL  PROG INDICATORS & SCOPE ROUTINE
                                ;THE BELOW TABLE CONTAINS ERROR INFORMATION NEEDED TO REPORT
                                ;MEMORY ERRORS DETECTED DURING PROGRAM RELOCATION. THE ERROR INFOR-
                                ;MATION IS PLACED IN THE TABLE BY THE 'SAVVAL' SUBROUTINE,AND
                                ;IS PROCESSED BY THE 'PNTREGS' SUBROUTINE.
000730  000000                  MEMTBL: .WORD   0                       ;CONTAINS 'GOOD' ADDRESS
000732  000000                          .WORD   0                       ;CONTAINS 'GOOD' DATA
000734  000000                          .WORD   0                       ;CONTAINS 'BAD' DATA
000736  000000                          .WORD   0                       ;CONTAINS 'BAD' DATA
000740  000000                  ECHO:   .WORD   0                       ;LOCATION FOR ECHOED CHARACTER
000742  020040  042440  051122  DEVERR: .ASCII  '   ERROR'
000750  051117
000752  005015      000         CRLF:   .ASCIZ  <15><12>
000755      134     000         SLASH:  .ASCIZ  '\'
000757      000                 DEV:    .BYTE   0                       ;CONTAINS DEVICE ID FOR ALL
000760      000                 IORETRY:.BYTE   0                       ;CONTAINS DEVICE RETRY COUNT
000761      000                 PEFLG:  .BYTE   0                       ;PARITY ERROR FLAG
                                        .EVEN
000762  000000                  EABITS: .WORD   0                       ;CONTAINS EA BITS FOR DISK XFERS
000764  000000                  STORE:  .WORD   0
000766  000400                  OPT.CP: .WORD   400                     ;CONTAINS OPTION AND CP INDICATORS
                                                                        ;EVEN BYTE:4=11/40,6=11/45
                                                                        ;ODD BYTE: 200=KT,100=EIS
                                                                        ;40=11/45 FPP,20=11/40 FIS
                                                                        ;10=STACK LIMIT (KJ)
                                                                        ;4=KW11-P,2=KW11-L,1=CONSOLE TTY

000770  000000                  OPTIONS:.WORD   0
000772      001                 PRDAT:  .BYTE   1                       ;CONTAINS NEXT DATA TO BE READ
000773      000                 PRSYNC: .BYTE   0                       ;CONTAINS SYNC COUNT
                        000770          .=770
000770      000                 MMON:   .BYTE   0               ;MEM MGMT ON/OFF IND 1/0=ON/OFF
000771      000                 QV:     .BYTE   0               ;QUICK VERIFY MODE IND
000772  000000                  DEVID:  .WORD   0               ;CONTAINS DEVICE INDENTIFIER
000774  000000                  LTICKS: .WORD   0               ;CONTAINS L CLOCK TICK COUNT
000776  000000                  PTICKS: .WORD   0               ;CONTAINS P CLOCK TICK COUNT
001000  000000                  ICNT:   .WORD   0               ;CONTAINS PASS COUNT
001002  001000                  $FILLS: .WORD   1000            ;CONTAINS FILLS COUNT (2) IN ODD BYTE
                                                                ;AND FILLER CHARACTER (0) IN EVEN BYTE
                                ;FILLER COUNTS: VT05 @2400 BD=4,VT05 @1200 BD=2,VT05 @600 BD=1
                                ;               LA30S @110 BD=2,LA30S @150 BD=4,LA30S @300 BD=12
                                ;               ALL VALUES ARE OCTAL

001004  000000                  FACTOR: .WORD   0               ;CONTAINS RELOCATION FACTOR, SUBTRACT # IN
                                                                ;FACTOR FROM PC TO GET PC OF ORIG CODE
001006  000000                  RELR1:  .WORD   0               ;CONTAINS RELOCATED R1 (THE R1 OF THE
                                                                ;ORIGINAL CODE MOVED)
001010  000000                  FRSTAD: .WORD   0               ;CONTAINS FIRST ADRS OF CODE TO BE MOVED
001012  000000                  FRSTMEM:.WORD   0               ;CONTAINS LOWER RELOCATION BOUNDARY ADDRESS

                                ;SCOPE (EMT) SERVICE ROUTINE
                                ;THIS ROUTINE ALLOWS THE SUBTEST TO BE CONTINUOUSLY LOOPED, ITERATED
                                ;(OR NOT ITERATED) BEFORE BEGINNING NEXT SUBTEST
001014  122737  000010  000766  $SCOPEA: CMPB    #10,@#OPT.CP
001022  001005                          BNE     10$
001024  005037  177766                  CLR     @#CPUERR
```

```
 1449  001030  012737  177777  177744            MOV     #-1,@#ERRREG
 1450  001036  032766  004000  000002  10$:      BIT     #4000,2(SP)     ;WAS REGISTER SET BIT SET
 1451  001044  001403                            BEQ     1$              ;BRANCH IF NOT
 1452  001046  052737  004000  177776            BIS     #4000,@#PSW     ;RETAIN REGISTER SET
 1453  001054  032737  040000  177570  1$:       BIT     #40000,@#SWR    ;CHECK BIT 14 (CONTINUOUS LOOP)
 1454  001062  001416                            BEQ     4$
 1455  001064  010116                  2$:       MOV     R1,(SP)         ;LOAD RETURN ADDRESS
 1456  001066  010137  001006                    MOV     R1,@#RELR1
 1457  001072  163737  001004  001006            SUB     @#FACTOR,@#RELR1;RELR1 CONTAINS UNRELOCATED R1
 1458  001100  032737  000400  177570            BIT     #400,@#SWR      ;LOAD PDP11/45 MICRO BREAK REG?
 1459  001106  001403                            BEQ     3$
 1460  001110  113737  177570  177770            MOVB    @#SWR,@#UBREAK  ;LOAD MICRO BREAK REG WITH SR0-7
 1461  001116  000002                  3$:       RTI                     ;RETURN TO SUBTEST
 1462  001120  032737  004000  177570  4$:       BIT     #4000,@#SWR     ;SUBTEST ITERATION DESIRED?
 1463  001126  001006                            BNE     7$              ;BRANCH IF NO ITERATION DESIRED?
 1464  001130  105337  001150                    DECB    @#ITCNT         ;DECREMENT SUBTEST ITERATION COUNT
 1465  001134  001353                            BNE     2$
 1466  001136  113737  001151  001150            MOVB    @#ITCNT+1,@#ITCNT       ;RESET ITERATION COUNT
 1467  001144  011601                  7$:       MOV     (SP),R1         ;GET ADDRESS OF NEXT TEST
 1468  001146  000746                            BR      2$
 1469  001150  000040                  ITCNT:    .WORD   40
```

```
1470                                                    .SBTTL  RELOC ROUTINE
1471                                            ;ROUTINE TO RELOCATE PROGRAM CODE
1472  001152  032737  010000  177570  RELOC:  BIT     #SW12,@#SWR     ;BRANCH IF SW12=0
1473  001160  001404                          BEQ     20$             ;SW12=1 & SW09=0 = NO RELOCATION
1474  001162  032737  001000  177570          BIT     #SW09,@#SWR     ;BRANCH IF SW09=0
1475  001170  001470                          BEQ     4$              ;NO RELOCATION IF SW12=1 & SW09=0
1476  001172  105737  000770          20$:    TSTB    @#MMON          ;BRANCH IF MEM MGMT IS ENABLED
1477  001176  001065                          BNE     4$              ;NO RELOCATION IF MEM MGMT IS ON
1478  001200  013700  001010                  MOV     @#FRSTAD,R0     ;GET FIRST ADDRESS OF CODE TO BE MOVED
1479  001204  010005                          MOV     R0,R5           ;SAVE
1480  001206  010204                          MOV     R2,R4           ;GET LAST ADDRESS OF CODE TO BE MOVED
1481  001210  160504                          SUB     R5,R4           ;R4 CONTAINS # OF BYTES TO RELOCATE
1482  001212  010203                          MOV     R2,R3           ;SAVE LAST ADDRESS OF CODE TO BE MOVED
1483  001214  005737  001004                  TST     @#FACTOR        ;FIRST RELOCATION IS TO ENDTAG+2
1484  001220  001004                          BNE     10$
1485  001222  010237  001366                  MOV     R2,@#RETPC      ;SAVE RETURN PC TO NEXT SECTION OF CODE
1486  001226  013702  001012                  MOV     @#FRSTMEM,R2    ;SET FIRST ADDDRESS
  87  001232  060204                  10$:    ADD     R2,R4           ;R4 CONTAINS LAST MEMORY ADDRESS
1488  001234  020437  005750                  CMP     R4,@#LSTMEM     ;EXIT IF INSUFFICIENT MEMORY
1489  001240  101051                          BHI     5$              ;AVAILABLE FOR RELOCATION
1490  001242  160204                          SUB     R2,R4           ;R4 NOW CONTAINS BYTE COUNT
1491  001244  005037  001004                  CLR     @#FACTOR        ;CLEAR RELOCATION FACTOR
1492  001250  105737  000771                  TSTB    @#QV
1493  001254  001013                          BNE     12$             ;CHECK FOR QV MODE
1494  001256  032737  000040  177570  11$:    BIT     #SW05,@#SWR     ;CHECK IF ALL DEVICES DESIRED FOR
1495  001264  001007                          BNE     12$             ;RELOCATION ROUND ROBIN STYLE
1496  001266  032737  000010  177570          BIT     #SW03,@#SWR     ;CHECK IF A DEVICE IS SPECIFIED
1497  001274  001410                          BEQ     1$
1498  001276  113737  177570  000757          MOVB    @#SWR,@#DEV     ;GET SELECTED DEVICE
1499  001304  005037  000762          12$:    CLR     @#EABITS        ;CLEAR EABITS FOR DEVICE
1500  001310  004767  000114                  JSR     PC,IODEV        ;GO RELOCATE VIA SELECTED DEVICE
1501  001314  102003                          BVC     2$              ;'V' =0/1 INDICATES NO ERROR/ERROR
1502  001316  012022                  1$:     MOV     (R0)+,(R2)+     ;RELOCATE PROGRAM CODE
1503  001320  020003                          CMP     R0,R3           ;CHECK IF DONE
1504  001322  001375                          BNE     1$
1505  001324  024042                  2$:     CMP     -(R0),-(R2)     ;CHECK THAT CODE WAS RELOCATED
1506  001326  001403                          BEQ     3$              ;PROPERLY
1507  001330  004767  001312                  JSR     PC,SAVVAL       ;GO SAVE PERTINENT DATA FOR TYPEOUT
1508  001334  104400                          HLT                     ;ERROR! CODE NOT RELOCATED PROPERLY
1509  001336  020005                  3$:     CMP     R0,R5           ;CHECK IF FINISHED CHECKING
1510  001340  001371                          BNE     2$
1511  001342  162737  000010  000772          SUB     #10,@#DEVID     ;BRANCH IF ERROR DETECTED ON RELOCATION
1512  001350  001742                          BEQ     11$
1513  001352  105237  000757          4$:     INCB    @#DEV           ;STEP TO NEXT DEVICE
1514  001356  005237  000772                  CLR     @#DEVID         ;SET DEVICE IND TO CP
1515  001362  010207                          MOV     R2,PC           ;GO EXECUTE RELOCATED CODE
1516  001364  011707                  5$:     MOV     (PC),PC         ;RETURN TO NEXT SECTION OF CODE
1517  001366  000000                  RETPC:  0                       ;CONTAINS PC OF NEXT SECTION OF CODE
1518
1519                                          ;WAIT LOOP FOR COMPLETION OF DEVICE TRANSFERS
1520  001370  013704                  WAITIO: MOV     @(PC)+,R4               ;GET CONTENTS OF DEVICE'S BUS
1521  001372  000000                  BUSADR: .WORD   0                       ;ADDRESS REGISTER
1522  001374  105737  000770                  TSTB    @#MMON                  ;BRANCH IF MEM MGMT IS NOT ON
1523  001400  001404                          BEQ     1$
1524  001402  042704  160000                  BIC     #160000,R4              ;CONVERT ADDRESS TO VIRTUAL ADRS
1525  001406  052704  040000                  BIS     #040000,R4
```

```
1526  001412  024414                1$:     CMP     -(R4),(R4)
1527  001414  001401                        BEQ     2$
1528  001416  104400                        HLT
1529  001420  062714  000000        2$:     ADD     #0,(R4)
1530  001424  000137                        JMP     @(PC)+                  ;GO TO WAITIO OR 41$ OR 71$ IN
1531  001426  001370                IODONE: .WORD   WAITIO                  ;IODEV ROUTINE BELOW
1532                                                                        ;;41$ WHEN WRITE COMPLETE
1533                                                                        ;71$ WHEN READ COMPLETE
```

```
1534                                    .SBTTL  IODEV ROUTINE
1535                                ;ROUTINE TO RELOCATE PROGRAM CODE VIA DEVICE SELECTED IN SWITCHES<2-0>
1536                                ;(IF SW03=1) OR VIA ALL DEVICES IF SW05=1.
1537                                ;THIS ROUTINE WRITES THE DATA TO BE RELOCATED ONTO THE SELECTED
1538                                ;DEVICE AND AFTER COMPLETION READS THE DATA BACK INTO MEMORY WHERE
1539                                ;THE RELOCTED DATA IS TO GO. AFTER THE READ THE ROUTINE RETURNS TO
1540                                ;THE CALLER. THE CALLER COMPARES THE DATA READ BACK.
1541                                ;DEVICES ARE:
1542                                ;       0-CP                            ;TAKES ERROR EXIT
1543                                ;       1-RK
1544                                ;       2-RF
1545                                ;       3-RP
1546                                ;       4-RC
1547                                ;       5-RP04/05/06
1548                                ;       6-RS04/03
1549                                ;       7-RK06
1550                                ;INPUT PARAMETERS:
1551                                ;       R0                              ;BUS ADDRESS FOR WRITE
1552                                ;       R1                              ;DON'T CARE
1553                                ;       R2                              ;BUS ADDRESS FOR READ
1554                                ;       R3                              ;DON'T CARE
1555                                ;       R4                              ;BYTE COUNT
1556                                ;       R5                              ;DON'T CARE
1557                                ;       EABITS                          ;LOADED
1558                                ;       DEV                             ;DEVICE IDENTIFIER
1559
1560                                ;OUTPUT
1561                                ;       R0                              ;UPDATED BY BYTE COUNT (IF NO ERROR)
1562                                ;       R1                              ;UNCHANGED
1563                                ;       R2                              ;UPDATED BY BYTE COUNT (IF NO ERROR)
1564                                ;       R3                              ;UNCHANGED
1565                                ;       R4                              ;CLOBBERED
1566                                ;       R5                              ;UNCHANGED
1567                                ;       EABITS                          ;UNCHANGED
1568                                ;       'V' BIT                         ;CLEAR/SET=NO ERROR/ERROR
1569
1570  001430  004767  001242        IODEV:  JSR     PC,CLRTBIT              ;CLEAR 'T' BIT & SAVE PSW
1571  001434  010546                  MOV     R5,-(SP)                ;SAVE R5 ON THE STACK
1572  001436  052737  000200  177776  BIS     #PRTY4,@#PSW            ;SET PRIORITY LEVEL 4
1573  001444  142737  000370  000757  1$:     BICB    #370,@#DEV      ;LIMIT DEVICE SELECT CODE
1574  001452  113705  000757          MOVB    @#DEV,R5                ;GET SELECTED DEVICE
1575  001456  006305                  ASL     R5                      ;FORM INDEX POINTER
1576  001460  016501  002626          MOV     VALDEV(R5),R1           ;GET VALID DEVICE TABLE
1577  001464  016505  002352          MOV     DEVTBL(R5),R5           ;GET SELECTED DEVICE TABLE
1578  001470  001005                  BNE     2$                      ;BRANCH IF I/O DEVICE SELECTED
1579
1580  001472                          99$:                            ;ERROR EXIT
1581  001472  004737  002722          JSR     PC,@#RESTPS             ;RESORE ORIGINAL PSW
1582  001476  012605                  MOV     (SP)+,R5                ;RESTORE R5
1583  001500  000262                  SEV                             ;SET 'V' BIT TO INDICATE FAILURE
1584  001502  000207          100$:   RTS     PC                      ;RETURN
1585
1586                                ;CHECK IF USER SELECTED DEVICE IS AVAILABLE
1587  001504  012737  000006  000004  2$:     MOV     #ERRVEC+2,@#ERRVEC      ;SET TIME OUT TRAP VECTOR
1588  001512  000261                  SEC                             ;SET 'C' IN PSW
1589  001514  005775  000010          TST     @10(R5)                 ;REFERENCE A DEVICE REG
```

```
1590  001520  012737  005540  000004          MOV     #ERPRT,@#ERRVEC         ;RESTORE ERROR TRAP VECTOR
1591                                                                          ;NOTE: MOV DOES NOT AFFECT 'C'
1592  001526  103402                          BCS     18$                     ;IF NOT THERE GO TO NEXT DEVICE
1593  001530  005701                          TST     R1                      ;TEST IF DEVICE VALID
1594  001532  001023                          BNE     20$                     ;EVER ONWARD IF VALID
1595  001534  105737  000771          18$:    TSTB    @#0V
1596  001540  001004                          BNE     15$
1597  001542  032737  000040  177570          BIT     #SW05,@#SWR             ;DO NOT REPORT ERROR IF ALL
1598  001550  001403                          BEQ     14$
1599  001552  105237  000757          15$:    INCB    @#DEV                   ;STEP TO NEXT DEVICE
1600  001556  000732                          BR      1$
1601  001560  113705  000757          14$:    MOVB    @#DEV,R5                ;GET DEVICE ID
1602  001564  006305                          ASL     R5                      ;FORM INDEX VALUE
1603  001566  016567  004422  031170          MOV     DEVICE(R5),DEVNAM       ;GET HIS NAME
1604  001574  000004  032762                  TYPE,NODEV
1605  001600  000734                          BR      99$                     ;TAKE ERROR EXIT
1606
1607  001602  012737  001370  001426  20$:    MOV     #WAITIO,@#IODONE        ;SET IODONE 'JMP' TO WAITIO
1608  001610  112737  000003  000760          MOVB    #3,@#IORETRY            ;SET ERROR RETRY COUNT
1609  001616  010427                          MOV     R4,(PC)+                ;SAVE BYTE COUNT
1610  001620  000000          10$:    .WORD   0
1611  001622  010446                          MOV     R4,-(SP)                ;FORM TWO'S COMPLEMENT
1612  001624  006216                          ASR     (SP)                    ;WORD COUNT
1613  001626  005416                          NEG     (SP)
1614  001630  012667  000344                  MOV     (SP)+,9$                ;AND SAVE IN 9$ BELOW
1615  001634  113737  000757  000772          MOVB    @#DEV,@#DEVID           ;SET DEVICE IDENT
1616  001642  013727  000762                  MOV     @#EABITS,(PC)+          ;SAVE EABITS IN 11$ BELOW
1617  001646  000000          11$:    .WORD   0
1618  001650  123727  000757  000004          CMPB    @#DEV,#4                ;BRANCH IF DEVICE IS NOT
1619  001656  003414                          BLE     12$                     ;A MASS BUS DEVICE
1620  001660  006367  177762                  ASL     11$                     ;SHIFT EA BITS TO
1621  001664  006367  177756                  ASL     11$                     ;POSITION 8-9
1622  001670  006367  177752                  ASL     11$                     ;FROM 4-5
1623  001674  006367  177746                  ASL     11$
1624  001700  012535                          MOV     (R5)+,@(R5)+            ;DO A READ IN (TO SET VOL VALID)
1625  001702  012735  010000                  MOV     #10000,@(R5)+           ;SET PDP11 FORMAT (IN RPOF REG)
1626  001706  012535                          MOV     (R5)+,@(R5)+            ;LOAD DEVICES UNIT #
1627  001710  012546          12$:    MOV     (R5)+,-(SP)             ;GET DEVICE'S VECTOR ADDRESS
1628  001712  012776  001766  000000          MOV     #4$,@(SP)               ;LOAD VECTOR
1629  001720  062716  000002                  ADD     #2,(SP)
1630  001724  012736  000240                  MOV     #PRTY5,@(SP)+           ;AND PSW ON INTERRUPT
1631  001730  004767  000246                  JSR     PC,DSKADR               ;GO GET RANDOM DISK ADDRESS
1632  001734  016735  000340          3$:     MOV     CYLADR,@(R5)+           ;SET 'CYLINDER' ADDRESS
1633  001740  016735  000342                  MOV     TRKSEC,@(R5)+           ;SET 'TRACK/SECTOR' ADDRESS
1634  001744  011537  001372                  MOV     (R5),@#BUSADR           ;SAVE ADDRESS OF BUS ADDRESS REG
1635  001750  010035                          MOV     R0,@(R5)+               ;SET BUS ADDRESS
1636  001752  016735  000222                  MOV     9$,@(R5)+               ;SET WORD COUNT
1637  001756  016535  000002                  MOV     2(R5),@(R5)+            ;AND SET COMMAND
1638  001762  000167  177402          30$:    JMP     WAITIO                  ;GO WAIT FOR WRITE TO FINISH
1639
1640  001766  012737  001776  001426  4$:     MOV     #41$,@#IODONE           ;SET RETURN FROM WAITIO ROUTINE
1641  001774  000002                          RTI
1642
1643  001776  012737  001370  001426  41$:    MOV     #WAITIO,@#IODONE
1644  002004  015504                          MOV     @-(R5),R4               ;GET AND CHECK ERROR BIT
1645  002006  100012                          BPL     5$                      ;BRANCH IF NO ERROR
```

```
1646  002010  104400                        HLT                     ;REPORT ERROR
1647  002012  016535  000006                MOV     6(R5),@(R5)+    ;RESET DEVICE'S CONTROLLER
1648  002016  162705  000012                SUB     #12,R5          ;RESET TABLE POINTER
1649  002022  105337  000760                DECB    @#IORETRY       ;RETRY WRITE COMMAND
1650  002026  001342                        BNE     3$
1651  002030  000167  177436        40$:    JMP     99$             ;TAKE ERROR EXIT
1652                                                                ;AFTER THREE RETRYS
1653
1654  002034  112737  000003  000760  5$:     MOVB    #3,@#IORETRY    ;RESET ERROR RETRY COUNT
1655  002042  162705  000012          6$:     SUB     #12,R5          ;RESET TABLE POINTER
1656  002046  012735  002114                  MOV     #7$,@(R5)+      ;RESET DEVICE'S INT VECTOR
1657  002052  016735  000222                  MOV     CYLADR,@(R5)+   ;GET 'CYLINDER' ADDRESS
1658  002056  016735  000224                  MOV     TRKSEC,@(R5)+   ;GET 'TRACK/SECTOR' ADDRESS
1659  002062  011537  001372                  MOV     (R5),@#BUSADR   ;SAVE ADDRESS OF BUS ADDRESS REG
1660  002066  010235                          MOV     R2,@(R5)+       ;SET BUS ADDRESS
1661  002070  016735  000104                  MOV     9$,@(R5)+       ;SET WORD COUNT
1662  002074  016746  177546                  MOV     11$,-(SP)       ;GET EA BITS
1663  002100  056516  000004                  BIS     4(R5),(SP)      ;SET IN READ COMMAND
1664  002104  012675  000000                  MOV     (SP)+,@(R5)     ;LOAD COMMAND
1665  002110  000240                          NOP
1666  002112  000723                          BR      30$             ;GO TO WAITIO VIA 30$
1667
1668                                ;WHEN READ IS FINISHED INTERRUPT TO HERE
1669  002114  012737  002124  001426  7$:     MOV     #71$,@#IODONE   ;SET IODONE 'JMP' TO 71$ BELOW
1670  002122  000002                          RTI
1671
1672  002124  012737  001370  001426  71$:    MOV     #WAITIO,@#IODONE ;RESET IODONE 'JMP' TO WAITIO
1673  002132  013504                          MOV     @(R5)+,R4       ;GET & CHECK ERROR BIT IN COMMAND REG
1674  002134  100007                          BPL     8$              ;BRANCH IF NO ERROR
1675  002136  104400                          HLT                     ;REPORT ERROR
1676  002140  016555  000004                  MOV     4(R5),@-(R5)    ;RESET DEVICE'S CONTROLLER
1677  002144  105337  000760                  DECB    @#IORETRY       ;RETRY READ COMMAND
1678  002150  001334                          BNE     6$              ;3 TIMES AND IF STILL FAILS
1679  002152  000726                          BR      40$             ;TAKE ERROR EXIT
1680  002154  012605                  8$:     MOV     (SP)+,R5        ;RESTORE R5
1681  002156  066700  177436                  ADD     10$,R0          ;ADD BYTE COUNT TO WRITE AND
1682  002162  066702  177432                  ADD     10$,R2          ;READ ADDRESSES (FOR CHECKING)
1683  002166  004767  000530                  JSR     PC,RESTPS       ;GO RESTORE 'T' IN PSW
1684  002172  000242                          CLV                     ;CLEAR ERROR INDICATOR
1685  002174  000167  177302                  JMP     100$            ;EXIT
1686
1687  002200  000000                  9$:     .WORD   0               ;CONTAINS TWO'S COMP WORD COUNT
1688
1689                                ;SUBROUTINE TO GENERATE RANDOM DSK SURFACE ADDRESSES
1690  002202  105737  000771        DSKADR: TSTB    @#QV
1691  002206  001004                          BNE     1$
1692  002210  032737  000020  177570          BIT     #20,@#SWR       ;BRANCH IF USER DOES NOT WANT
1693  002216  001426                          BEQ     2$              ;RANDOM DISK ADDRESSES
1694  002220  010046                  1$:     MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
1695  002222  013700  000772                  MOV     @#DEVID,R0      ;GET I/O DEVICE ID
1696  002226  006300                          ASL     R0              ;FORM INDEX INTO
1697  002230  006300                          ASL     R0              ;ADRTAB BELOW
1698  002232  060146                          ADD     R1,-(SP)        ;FORM RANDOM #
1699  002234  005516                          ADC     (SP)
1700  002236  011667  000036                  MOV     (SP),CYLADR     ;MOVE TO 'CYLINDER' ADDRESS
1701  002242  046067  002312  000030          BIC     ADRTAB(R0),CYLADR ;LIMIT 'CYLINDER' ADDRESS
```

```
1702  002250  060116                          ADD     R1,(SP)
1703  002252  005516                          ADC     (SP)
1704  002254  012667  000026                  MOV     (SP)+,TRKSEC            ;MOVE TO 'TRACK/SECTOR' ADDRESS
1705  002260  005720                          TST     (R0)+
1706  002262  046067  002312  000016          BIC     ADRTAB(R0),TRKSEC               ;LIMIT 'TRACK/SEC' ADRS
1707  002270  012600                          MOV     (SP)+,R0                ;RESTORE R0
1708  002272  000207                          RTS     PC                      ;RETURN
1709  002274  012727  000000          2$:     MOV     #0,(PC)+                ;SET CYLINDER ADDRESS = 0
1710  002300  000000          CYLADR: .WORD   0
1711  002302  012727  000000                  MOV     #0,(PC)+                ;SET TRACK & SECTOR = 0
1712  002306  000000          TRKSEC: .WORD   0
1713  002310  000207                          RTS     PC
1714
1715                                  ;TABLE OF DEVICE 'CYLINDER' AND 'TRACK/SECTOR' ADDRESS LIMITERS
1716  002312  000000          ADRTAB: .WORD   0                               ;NOT USED
1717  002314  000000                  .WORD   0                               ;NOT USED
1718  002316  163350                  .WORD   163350                          ;RKDA LIMITER
1719  002320  163350                  .WORD   163350                          ;RKDA LIMITER
1720  002322  177774                  .WORD   177774                          ;RFDAE LIMITER
1721  002324  020000                  .WORD   020000                          ;RFDAR LIMITER
1722  002326  177152                  .WORD   177152                          ;RPCA LIMITER
1723  002330  170370                  .WORD   170370                          ;RPDA LIMITER
1724  002332  176400                  .WORD   176400                          ;RCDA LIMITER
1725  002334  176400                  .WORD   176400                          ;RCDA LIMITER
1726  002336  177145                  .WORD   177145                          ;RP4CA LIMITER
1727  002340  170370                  .WORD   170370                          ;RP4DST LIMITER
1728  002342  170400                  .WORD   170400                          ;RSDA LIMITER
1729  002344  170400                  .WORD   170400                          ;RSDA LIMITER
1730  002346  177400                  .WORD   177400                          ;RK6DC LIMITER
1731  002350  173014                  .WORD   173014                          ;RK6DA LIMITER
1732
1733                                  .SBTTL  DEVICE TABLES
1734  002352  000000          DEVTBL: .WORD   0
1735  002354  002372                  .WORD   RKTBL
1736  002356  002414                  .WORD   RFTBL
1737  002360  002436                  .WORD   RPTBL
1738  002362  002460                  .WORD   RCTBL
1739  002364  002502                  .WORD   RP4TBL                          ;RESERVED FOR RP04
1740  002366  002536                  .WORD   RSTBL
1741  002370  002572                  .WORD   RK6TBL
1742
1743  002372  000220          RKTBL:  .WORD   RKVEC
1744  002374  177412                  .WORD   RKDA
1745  002376  177412                  .WORD   RKDA
1746  002400  177410                  .WORD   RKBA
1747  002402  177406                  .WORD   RKWC
1748  002404  177404                  .WORD   RKCS
1749  002406  000503                  .WORD   503                             ;WRITE COMMAND
1750  002410  000505                  .WORD   505                             ;READ COMMAND
1751  002412  000001                  .WORD   1                               ;CONTROL RESET
1752
1753  002414  000204          RFTBL:  .WORD   RFVEC
1754  002416  177470                  .WORD   RFDAE
1755  002420  177466                  .WORD   RFDAR
1756  002422  177464                  .WORD   RFCMA
1757  002424  177462                  .WORD   RFWC
```

```
1758  002426  177460                  .WORD   RFDCS
1759  002430  000103                  .WORD   103             ;WRITE COMMAND
1760  002432  000105                  .WORD   105             ;READ COMMAND
1761  002434  000001                  .WORD   1               ;CONTROL RESET
1762
1763  002436  000254          RPTBL:  .WORD   RPVEC
1764  002440  176722                  .WORD   RPCA
1765  002442  176724                  .WORD   RPDA
1766  002444  176720                  .WORD   RPBA
1767  002446  176716                  .WORD   RPWC
1768  002450  176714                  .WORD   RPCS
1769  002452  000103                  .WORD   103             ;WRIE COMMAND
1770  002454  000105                  .WORD   105             ;READ COMMAND
1771  002456  000001                  .WORD   1               ;CONTROL RESET
1772
1773  002460  000210          RCTBL:  .WORD   RCVEC
1774  002462  177442                  .WORD   RCDA
1775  002464  177442                  .WORD   RCDA
1776  002466  177452                  .WORD   RCCA
1777  002470  177450                  .WORD   RCWC
1778  002472  177446                  .WORD   RCCS
1779  002474  000103                  .WORD   103             ;WRITE COMMAND
1780  002476  000105                  .WORD   105             ;READ COMMAND
1781  002500  000001                  .WORD   1               ;CONTROL RESET
1782
1783                          ;RP04 TABLE
1784  002502  000023          RP4TBL: .WORD   23
1785  002504  176700                  .WORD   RP4CS1
1786  002506  176732                  .WORD   RP40F
1787  002510  000000                  .WORD   0               ;RP04 UNIT #
1788  002512  176710                  .WORD   RP4CS2          ;RP4CS2 REGISTER ADDRESS
1789  002514  000254                  .WORD   RP4VEC
1790  002516  176734                  .WORD   RP4CA
1791  002520  176706                  .WORD   RP4DST
1792  002522  176704                  .WORD   RP4BA
1793  002524  176702                  .WORD   RP4WC
1794  002526  176700                  .WORD   RP4CS1
1795  002530  000161                  .WORD   161             ;WRITE COMMAND
1796  002532  000171                  .WORD   171             ;READ COMMAND
1797  002534  040011                  .WORD   40011           ;DRIVE CLEAR
1798
1799                          ;RS04 TABLE
1800  002536  000021          RSTBL:  .WORD   21
1801  002540  172040                  .WORD   RSCS1
1802  002542  172040                  .WORD   RSCS1
1803  002544  000000                  .WORD   0               ;RS04 UNIT #
1804  002546  172050                  .WORD   RSCS2           ;RSCS2 REGISTER ADDRESS
1805  002550  000204                  .WORD   RSVEC
1806  002552  172046                  .WORD   RSDA
1807  002554  172046                  .WORD   RSDA
1808  002556  172044                  .WORD   RSBA
1809  002560  172042                  .WORD   RSWC
1810  002562  172040                  .WORD   RSCS1
1811  002564  000161                  .WORD   161
1812  002566  000171                  .WORD   171
1813  002570  040011                  .WORD   40011
```

```
                                ;RK06 TABLE
002572  000003                  RK6TBL: .WORD   3
002574  177440                          .WORD   RK6CS1
002576  177456                          .WORD   RK6OF
002600  000000                          .WORD   0
002602  177450                          .WORD   RK6CS2
002604  000210                          .WORD   RK6VEC
002606  177460                          .WORD   RK6DC
002610  177446                          .WORD   RK6DA
002612  177444                          .WORD   RK6BA
002614  177442                          .WORD   RK6WC
002616  177440                          .WORD   RK6CS1
002620  000023                          .WORD   23
002622  000021                          .WORD   21
002624  000005                          .WORD   5

                                ;TABLE OF DEVISES PRESENT AT RUN TIME,THE LOCATION WILL BE
                                ;ZEROED IF DEVICE WAS NOT THERE
002626  000000                  VALDEV: .WORD   0                       ;CPU
002630  177400                          .WORD   177400                  ;RK05
002632  177460                          .WORD   177460                  ;RF11
002634  176714                          .WORD   176714                  ;RP11
002636  177440                          .WORD   177440                  ;RC11
002640  176700                          .WORD   176700                  ;RP04/05/06
002642  172040                          .WORD   172040                  ;RS04/03
002644  177440                          .WORD   177440                  ;RK06

                                ;ROUTINE TO SAVE MEMORY VALUES ON RELOCATION ERROR
002646  012737  000010  000772  SAVVAL: MOV     #10,@#DEVID             ;SET DEVICE IND=MEMORY
002654  010446                          MOV     R4,-(SP)                ;SAVE R4 ON THE STACK
002656  012704  000730                  MOV     #MEMTBL,R4              ;GET STARTING ADDRESS OF TABLE
002662  010024                          MOV     R0,(R4)+                ;LOAD 'GOOD' ADDRESS
003664  011024                          MOV     (R0),(R4)+              ;LOAD 'GOOD' DATA
002666  010224                          MOV     R2,(R4)+                ;LOAD 'BAD' ADDRESS
002670  011224                          MOV     (R2),(R4)+              ;LOAD 'BAD' DATA
002672  012604                          MOV     (SP)+,R4                ;RESTORE R4
002674  000207                          RTS     PC                      ;EXIT

                                ;ROUTINE TO CLEAR 'T' BIT
002676  013746  177776          CLRTBIT:MOV     @#PSW,-(SP)     ;PUSH PSW ONTO STACK
002702  011627                          MOV     (SP),(PC)+      ;SAVE IN RETPSW BELOW
002704  000000                  RETPSW: .WORD   0
002706  042716  000020                  BIC     #20,(SP)        ;CLEAR 'T' BIT IN PSW ON STACK
002712  012746  002720          RESPSW: MOV     #1$,-(SP)       ;SET RETURN PC FOR RTI
002716  000002                          RTI                     ;CLEAR 'T' BIT IN PSW
002720  000207                  1$:     RTS     PC              ;RETURN

002722  042737  177400  177776  RESTPS: BIC     #177400,@#PSW   ;SET KERNEL MODE
002730  016746  177750                  MOV     RETPSW,-(SP)    ;PUSH ORIG PSW ONTO STACK
002734  000766                          BR      RESPSW
```

```
1865                                                    .SBTTL  TYPE SUBROUTINE
1866                                            ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1867  002736  010046                            .TYPE:  MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
1868  002740  017600  000002                           MOV     @2(SP),R0       ;GET MESSAGE ADDRESS
1869  002744  062766  000002  000002                   ADD     #2,2(SP)        ;ADJUST RETURN PC
1870  002752  032737  000400  000766                   BIT     #TTOPT,@#OPT.CP ;BRANCH IF NO CONSOLE TTY AVAILABLE
1871  002760  001410                                   BEQ     6$
1872  002762  005767  003270                           TST     NOTYPE          ;BRANCH IF NO TYPING DESIRED (VIA ↑O)
1873  002766  001005                                   BNE     6$
1874
1875  002770  112046                            1$:     MOVB    (R0)+,-(SP)     ;PUSH CHAR ON THE STACK
1876  002772  001005                                   BNE     2$              ;BRANCH IF NOT TERMINATOR
1877  002774  004767  000040                           JSR     PC,5$           ;TYPE NULL CHARACTER
1878  003000  005726                                   TST     (SP)+           ;POP TERMINATOR OFF THE STACK
1879  003002  012600                            6$:     MOV     (SP)+,R0        ;RESTORE R0
1880  003004  000002                                   RTI                     ;RETURN
1881
1882  003006  004767  000026                    2$:     JSR     PC,5$           ;TYPE CHARACTER
1883  003012  122726  000012                    3$:     CMPB    #12,(SP)+       ;CHECK IF CHAR WAS A LINE FEED
1884  003016  001364                                   BNE     1$              ;BRANCH IF NOT LINE FEED
1885
1886  003020  016746  175756                           MOV     $FILLS,-(SP)    ;GET # OF FILLERS REQUIRED AFTER
1887                                                                            ;LINE FEED AND FILLER CHARACTER
1888  003024  105366  000001                    4$:     DECB    1(SP)           ;DECREMENT FILLERS COUNT
1889  003030  002770                                   BLT     3$              ;BRANCH IF NO MORE FILLERS NEEDED
1890  003032  004767  000002                           JSR     PC,5$           ;TYPE FILLER CHARACTER
1891  003036  000772                                   BR      4$
1892
1893  003040  105737  177564                    5$:     TSTB    @#TPS           ;WAIT FOR OUTPUT DEVICE
1894  003044  100375                                   BPL     .-4             ;TO BECOME READY
1895  003046  116637  000002  177566                   MOVB    2(SP),@#TPB     ;TYPE CHARACTER
1896  003054  000207                                   RTS     PC
1897
1898          000000                                   NULL=0
1899
1900                                            ;SUBROUTINE TO CONVERT 16 BIT DATA TO ASCIZ STRING. THE ASCIZ STRING
1901                                            ;STARTS AT DIGITS AND IS 8 BYTES LONG. 6 ASCII DIGITS + 'SPACE' + '0'.
1902
1903  003056                                    CNVDAT:
1904  003056  004767  002306                           JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
1905  003062  012704  003266                           MOV     #DIGBUF+8.,R4           ;SET ADDRESS OF DIGIT BUFFER
1906  003066  010201                                   MOV     R2,R1                   ;GET DATA
1907  003070  005003                                   CLR     R3
1908  003072  012700  000006                           MOV     #6.,R0                  ;SET DIGIT COUNT
1909  003076  000167  000100                           JMP     CNVDIG                  ;GO TO DIGIT CONVERSION ROUTINE
1910
1911                                            ;SUBROUTINE TO CONVERT A VIRTUAL ADDRESS TO AN ASIZ STRING PHYSICAL
1912                                            ;ADDRESS. THE CONVERTED ASCIZ STRING IS AT 'DIGBUF' AND IS 10 BYTES LONG
1913                                            ;(8 DIGITS + 1 SPACE + 0 BYTE)
1914                                            ;CALL:  MOV     ADDRESS,R1              ;GET ADDRESS
1915                                            ;       JSR     PC,CNVADR
1916                                            ;NOTE: SUBROUTINE SUBRACTS 2 FROM ADDRESS BEFORE CONVERSION
1917                                            ;FOR EXAMPLE TO TYPE ERROR PC
1918                                            ;       MOV     PC,R1           ;IT IS THE PC OF THE MOV
1919                                            ;       JSR     PC,CNVADR       ;THAT GETS TYPED
1920                                            ;       TYPE
```

```
1921                                    ;       DIGBUF
1922
1923  003102                            CNVADR:
1924  003102  004767  002262                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
1925  003106  012704  003266                  MOV     #DIGBUF+8.,R4           ;GET ADDRESS OF DIGIT BUFFER
1926  003112  162701  000002                  SUB     #2,R1           ;SUBTRACT 2 FROM ADDRESS
1927  003116  010105                          MOV     R1,R5           ;SAVE ADDRESS TO BE CONVERTED
1928  003120  005003                          CLR     R3
1929  003122  105737  000770                  TSTB    @#MMON          ;BRANCH IF MEM MGMT IS DISABLED
1930  003126  001423                          BEQ     3$
1931  003130  042701  017777                  BIC     #17777,R1               ;CLEAR ALL BUT PAR SELECTOR BITS
1932  003134  006301                          ASL     R1              ;SHIFT BITS 15-13 OF ADDRESS
1933  003136  006101                          ROL     R1              ;LEFT TO
1934  003140  006101                          ROL     R1              ;3-1
1935  003142  006101                          ROL     R1
1936  003144  006301                          ASL     R1
1937  003146  062701  172340                  ADD     #KIPAR0,R1              ;FORM ADDRESS OF PAR REG
1938  003152  011101                          MOV     (R1),R1         ;GET CONTENTS OF PAR
1939  003154  012700  000006                  MOV     #6,R0           ;SET SHIFT COUNTER
1940  003160  006301                  2$:     ASL     R1              ;SHIFT PAR BITS IN R1
1941  003162  006103                          ROL     R3              ;6 PLACES LEFT TO R3-R1
1942  003164  077003                          SOB     R0,2$
1943  003166  042705  160000                  BIC     #160000,R5              ;CLEAR PAR SELECTOR BITS IN ADDRESS
1944  003172  060501                          ADD     R5,R1           ;FORM PHYSICAL ADDRESS
1945  003174  005503                          ADC     R3              ;IN R1 & R3
1946  003176  012700  000010          3$:     MOV     #8.,R0          ;SET DIGIT COUNT
1947
1948  003202  012705  000003          CNVDIG: MOV     #3,R5           ;AND BITS PER DIGIT COUNT
1949  003206  005002                          CLR     R2              ;R2 WILL CONTAIN DIGIT
1950  003210  006203                  5$:     ASR     R3              ;R3<00> TO 'C'
1951  003212  006001                          ROR     R1              ;'C' TO R1<15> & R1<00> TO 'C'
1952  003214  106002                          RORB    R2              ;'C' TO R2<07>
1953  003216  005305                          DEC     R5              ;DECREMENT SHIFT COUNT
1954  003220  001373                          BNE     5$
1955  003222  012705  000005                  MOV     #5,R5           ;SET SHIFT COUNT
1956  003226  000241                  6$:     CLC
1957  003230  106002                          RORB    R2              ;SHIFT DIGIT FROM <07-05>
1958  003232  005305                          DEC     R5              ;TO <02-00>
1959  003234  001374                          BNE     6$
1960
1961  003236  062702  000260                  ADD     #260,R2         ;CONVERT DIGIT TO ASCII
1962  003242  110244                          MOVB    R2,-(R4)        ;MOVE DIGIT INTO DIGIT BUFFER
1963  003244  005300                          DEC     R0              ;DECREMENT DEGIT COUNT
1964  003246  001355                          BNE     CNVDIG          ;CONVERT NEXT DIGIT
1965  003250  004767  002234                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
1966  003254  000207                          RTS     PC
1967
1968                                    ;DIGIT BUFFER
1969  003256      000     000           DIGBUF: .BYTE   0,0
1970  003260  000006                   DIGITS: .BLKB   6.
1971  003266      040                           .BYTE   40              ;'SPACE'
1972  003267      000                           .BYTE   0               ;'0' TERMINATOR
1973
1974                                    ;SUBROUTINE TO CONVERT 16 BIT OCTAL DATA TO AN ASCIZ STRING AND TYPE IT.
1975                                    ;CALL:  MOV     #DATA,R2                ;LOAD R2 WITH THE DATA
1976                                    ;       JSR     PC,TYPDAT
```

```
                                ;SUBROUTINE TO CONVERT A VIRTUAL ADDRESS TO A PHYSICAL ADDRESS AND TYPE IT.
                                ;CALL:  MOV     #ADDRESS,R1             ;LOAD R1 WITH THE ADDRESS
                                ;       JSR     PC,TYPADR
                                
003302  004767  177574        TYPADR: JSR     PC,CNVADR               ;CONVERT ADDRESS TO ASCIZ STRING
003306  000004  003256                TYPE,DIGBUF                     ;TYPE ADDRESS
003312  000207                        RTS     PC
                                
                                ;KEYBOARD INTERRUPT SERVICE ROUTINE
                                
                        000003  CNTRLC=3
                        000017  CNTRLO=17
                                
003314  000240          TKISR:  NOP
003316  013746  177562          MOV     @#TKB,-(SP)             ;GET CHARACTER
003322  042716  177600          BIC     #177600,(SP)            ;STRIP UNUSED BITS
003326  022716  000003          CMP     #CNTRLC,(SP)            ;BRANCH IF NOT CONTROL C (↑C)
003332  001005                  BNE     1$
003334  000004  000752          TYPE,CRLF                       ;ECHO <CR><LF>
003340  005726                  TST     (SP)+                   ;POP CHARACTER OFF THE STACK
003342  000000                  HALT
003344  000002                  RTI                             ;RETURN
                                
003346  122716  000015  1$:     CMPB    #15,(SP)                ;BRANCH IF NOT <CR>
003352  001004                  BNE     2$
003354  000004  000752          TYPE,CRLF                       ;ECHO <CR><LF>
003360  005726                  TST     (SP)+                   ;POP CHARACTEROFF STACK
003362  000002                  RTI                             ;RETURN
                                
003364  122716  000017  2$:     CMPB    #CNTRLO,(SP)            ;BRANCH IF NOT CONTROL O (↑O)
003370  001005                  BNE     3$
003372  005167  002660          COM     NOTYPE
003376  112716  000015          MOVB    #15,(SP)                ;TYPE <CR><LF>
003402  000761                  BR      1$
                                
003404  112667  175330  3$:     MOVB    (SP)+,ECHO
003410  000004  000740          TYPE,ECHO                       ;ECHO CHARACTER
003414  000002                  RTI                             ;RETURN
```

```
                                                .SBTTL  ERROR SERVICE ROUTINE
                                        ;ERROR SERVICE CALLED BY TRAP (HLT) INSTRUCTION
003416  005737  177570          .HLT:   TST     @#SWR           ;HALT ON ERROR?
003422  100001                  BPL     .+4
003424  000000                  HALT                            ;ERROR PC IS TOP WORD ON STACK
003426  032737  020000  177570  BIT     #20000,@#SWR    ;TYPE OUT DESIRED?
003434  001117                  BNE     1$              ;BRANCH IF NO TYPEOUT
003436  004767  001726          JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
003442  013702  001000          MOV     @#ICNT,R2       ;GET PASS COUNT
003446  004767  177404          JSR     PC,CNVDAT
003452  016767  177604  000330  MOV     DIGITS+2,PASSES ;LOAD ASCII VALUES
003460  016767  177600  000324  MOV     DIGITS+4,PASSES+2
003466  000004  004000          TYPE,PASCNT
003472  016602  000016          MOV     16(SP),R2       ;GET PC OF ERROR CALL
003476  124242                  CMPB    -(R2),-(R2)     ;DECREMENT PC TO HLT
003500  000004  004015          TYPE,VIRPC
003504  004767  177560          JSR     PC,TYPDAT       ;TYPE DATA
003510  016702  175256          MOV     DEVID,R2        ;GET DEVICE IDENTIFICATION
003514  001411                  BEQ     13$             ;AND BRANCH IF DEVICE WAS CP
003516  006302                  ASL     R2
003520  016267  004422  175214  MOV     DEVICE(R2),DEVERR
003526  000004  000742          TYPE,DEVERR
003532  004767  000434          JSR     PC,PNTREGS
003536  000454                  BR      19$
003540  000004  004023  13$:    TYPE,STATUS
003544  016602  000020          MOV     20(SP),R2       ;GET STATUS AT TIME OF ERROR
003550  004767  177514          JSR     PC,TYPDAT       ;TYPE STATUS
003554  122737  000010  000766  CMPB    #10,@#OPT.CP
003562  001014                  BNE     12$
003564  000004  004030          TYPE,CPERR
003570  013702  177766          MOV     @#CPUERR,R2
003574  004767  177470          JSR     PC,TYPDAT
003600  000004  004035          TYPE,ERREG
003604  013702  177744          MOV     @#ERRREG,R2
003610  004767  177454          JSR     PC,TYPDAT
003614  016602  000016  12$:    MOV     16(SP),R2       ;GET PC OF ERROR
003620  124242                  CMPB    -(R2),-(R2)
003622  105737  000770          TSTB    @#MMON          ;CHECK IF MEM MGMT IS ENABLED
003626  001012                  BNE     10$             ;BRANCH IF ENABLED
003630  005737  001004          TST     @#FACTOR
003634  001415                  BEQ     19$
003636  000004  004042          TYPE,RELPC
003642  163702  001004          SUB     @#FACTOR,R2     ;FORM PC OF ORIGINAL CODE
003646  004767  177416          JSR     PC,TYPDAT       ;TYPE DATA
003652  000406                  BR      19$             ;GO TO 19$
003654  000004  004047  10$:    TYPE,PHYSPC
003660  016601  000016          MOV     16(SP),R1       ;GET ERROR PC
003664  004767  177412          JSR     PC,TYPADR       ;TYPE ADDRESS
003670                  19$:
003670  004767  001514          JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
003674  032737  002000  177570  1$:     BIT     #2000,@#SWR     ;RING BELL ON ERROR
003702  001402                  BEQ     2$
003704  000004  004054          TYPE,BELL
003710  005737  177570  2$:     TST     @#SWR           ;HALT AFTER TYPEOUT
003714  100001                  BPL     .+4
003716  000000                  HALT
```

```
2076  003720  005046                          CLR     -(SP)           ;ALLOW TIME FOR TTY TO TYPE CHAR
2077  003722  005316                  3$:     DEC     (SP)
2078  003724  001376                          BNE     3$
2079  003726  005737  005566                  TST     @#ERFLAG
2080  003732  001404                          BEQ     4$
2081  003734  005037  005566                  CLR     @#ERFLAG
2082  003740  005726                          TST     (SP)+
2083  003742  000406                          BR      6$              ;GO TO 6$
2084  003744  105737  000761          4$:     TSTB    @#PEFLG         ;BRANCH IF NO PARITY ERROR
2085  003750  001402                          BEQ     5$
2086  003752  000137  004646                  JMP     @#PERET         ;RETURN TO PARITY ERROR SERVICE
2087  003756  000002                  5$:     RTI
2088  003760  012716  003766          6$:     MOV     #7$,(SP)        ;GO TO 7$ AFTER RTI
2089  003764  000002                          RTI
2090  003766  000111                  7$:     JMP     (R1)            ;GO TO LAST 'SCOPE'
2091
2092                                  ;DIGIT TABLE
2093  003770  030460                  DIGTAB: "01
2094  003772  031462                          "23
2095  003774  032464                          "45
2096  003776  033466                          "67
2097  004000  005015  040520  051523  PASCNT: .ASCII  <15><12>'PASS #'
2098  004006  021440
2099
2100                                  ;NOTE:   PASSES MUST BE AT AN EVEN ADDRESS!
2101
2102  004010  030060  030060     000  PASSES: .ASCIZ  '0000'
2103  004015     040  050126  036503  VIRPC:  .ASCIZ  ' VPC='
2104  004022     000
2105  004023     120  053523  000075  STATUS: .ASCIZ  'PSW='
2106  004030  050103  036525     000  CPERR:  .ASCIZ  'CPU='
2107  004035     105  051122  000075  ERREG:  .ASCIZ  'ERR='
2108  004042  050122  036503     000  RELPC:  .ASCIZ  'RPC='
2109  004047     120  041520  000075  PHYSPC: .ASCIZ  'PPC='
2110  004054  000007                  BELL:   .ASCIZ  <7>
2111                                          .EVEN
2112  004056  005015                  SUCCESS:.ASCII  <15><12>
2113  004060  052040  042510  050440          .ASCII  / THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS BACK 0123456789 PASS# /
2114  004066  044525  045503  041040
2115  004074  047522  047127  043040
2116  004102  054117  045040  046525
2117  004110  051520  047440  042526
2118  004116  020122  044124  020105
2119  004124  040514  054532  042040
2120  004132  043517  020123  040502
2121  004140  045503  030040  031061
2122  004146  032063  033065  034067
2123  004154  020071  040520  051523
2124  004162  020043
2125  004164  030060  030060     000  PASSNO: .ASCIZ  '0000'
2126          004172                          .EVEN
2127                                  ;ROUTINE TO TYPE CONTENTS OF DEVICE REGISTER ON AN ERROR
2128                                  ;INPUT:
2129                                  ;       R2                              ;INDEX VALUE TO APPROPRIATE DEV
2130  004172  016200  004444          PNTREGS:MOV     REGS(2),R0              ;GET # OF REGS TO TYPE
2131  004176  016203  004466                  MOV     REGADR(2),R3            ;GET FIRST ADDRESS OF DATA TABLE
```

```
2132  004202  022703  000730            CMP     #MEMTBL,R3              ;BRANCH IF MEMORY ERROR
2133  004206  001421                    BEQ     2$
2134  004210  012302            1$:     MOV     (R3)+,R2
2135  004212  004767  177052            JSR     PC,TYPDAT               ;TYPE DATA
2136  004216  005267  000106            INC     COUNT
2137  004222  001005                    BNE     3$
2138  004224  000004  000752            TYPE,CRLF
2139  004230  012767  177766  000072    MOV     #177766,COUNT
2140  004236  005300            3$:     DEC     R0
2141  004240  001363                    BNE     1$
2142  004242  012767  177766  000060    MOV     #177766,COUNT
2143  004250  000207                    RTS     PC
2144
2145  004252  000004  004332    2$:     TYPE,GDADR
2146  004256  012301                    MOV     (R3)+,R1                ;GET 'FROM' ADDRESS
2147  004260  005721                    TST     (R1)+                   ;ADD 2
2148  004262  004767  177014            JSR     PC,TYPADR               ;TYPE ADDRESS
2149  004266  000004  004402            TYPE,A.DATA
2150  004272  012302                    MOV     (R3)+,R2                ;GET 'FROM' DATA
2151  004274  004767  176770            JSR     PC,TYPDAT               ;TYPE DATA
2152  004300  000004  004410            TYPE,BDADR
2153  004304  012301                    MOV     (R3)+,R1                ;GET 'TO' ADDRESS
2154  004306  005721                    TST     (R1)+                   ;ADD 2
2155  004310  004767  176766            JSR     PC,TYPADR               ;TYPE ADDRESS
2156  004314  000004  004402            TYPE,A.DATA
2157  004320  012302                    MOV     (R3)+,R2                ;GET 'TO' DATA
2158  004322  004767  176742            JSR     PC,TYPDAT               ;TYPE DATA
2159  004326  000207                    RTS     PC
2160  004330  177766            COUNT:  177766
2161
2162  004332  051105  047522  020122  GDADR:  .ASCII  'ERROR ON PROGRAM RELOCATION'<15><12>
2163  004340  047117  050040  047522
2164  004346  051107  046501  051040
2165  004354  046105  041517  052101
2166  004362  047511  006516     012
2167  004367     107  047517  020104          .ASCIZ  'GOOD ADRS='
2168  004374  042101  051522  000075
2169  004402  040504  040524  000075  A.DATA: .ASCIZ  'DATA='
2170  004410  040502  020104  042101  BDADR:  .ASCIZ  'BAD ADRS='
2171  004416  051522  000075
2172                                    .EVEN
2173
2174  004422  030060            DEVICE: .ASCII  'DD'
2175  004424  045522                    .ASCII  'RK'
2176  004426  043122                    .ASCII  'RF'
2177  004430  050122                    .ASCII  'RP'
2178  004432  041522                    .ASCII  'RC'
2179  004434  050122                    .ASCII  'RP'
2180  004436  051522                    .ASCII  'RS'
2181  004440  054130                    .ASCII  'XX'                    ;RESERVED FOR FUTURE USE
2182  004442  046515                    .ASCII  'MM'                    ;MEMORY
2183                            ;THE BELOW TABLE CONTAINS THE # OF DEVICE REGISTERS TO TYPE ON A
2184                            ;DEVICE ERROR
2185  004444  000001            REGS:   .WORD   1                       ;NOT USED (FOR CP)
2186  004446  000006                    .WORD   6                       ;TYPE 6 RK REGISTERS
2187  004450  000006                    .WORD   6                       ;TYPE 6 RF REGISTERS
```

```
2188  004452  000010                          .WORD   8.                      ;TYPE 8. RP REGISTERS
2189  004454  000006                          .WORD   6                       ;TYPE 6 RC REGISTERS
2190  004456  000024                          .WORD   20.                     ;TYPE 20. RP04 REGISTERS
2191  004460  000014                          .WORD   12.                     ;TYPE 12. RS REGS
2192  004462  000001                          .WORD   1
2193  004464  000004                          .WORD   4
2194
2195  004466  000000                  REGADR: .WORD   0
2196  004470  177400                          .WORD   RKDS
2197  004472  177460                          .WORD   RFDCS
2198  004474  176710                          .WORD   RPDS
2199  004476  177440                          .WORD   RCLA
2200  004500  176700                          .WORD   RP4CS1
2201  004502  172040                          .WORD   RSCS1
2202  004504  000000                          .WORD   0
2203  004506  000730                          .WORD   MEMTBL
2204
2205
2206                                  ;ROUTINE TO GET TYPED OCTAL ADDRESS AND CONVERT TO OCTAL. CALL:
2207                                  ;       JSR     R5,RECO
2208                                  ;       .WORD   0               ;CONVERTED DATA IS PLACED HERE
2209  004510  010046                  RECO:   MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
2210  004512  005015                          CLR     (R5)            ;CLEAR OLD DATA
2211  004514  105737  177560          1$:     TSTB    @#TKS           ;WAIT FOR USER TO INPUT CHARACTER
2212  004520  100375                          BPL     1$
2213  004522  113700  177562                  MOVB    @#TKB,R0        ;GET CHARACTER
2214  004526  042700  000200                  BIC     #200,R0         ;STRIP MSB
2215  004532  122700  000177                  CMPB    #177,R0         ;CHECK IF RUBOUT
2216  004536  001007                          BNE     2$              ;BRANCH IF NOT RUBOUT
2217  004540  000004  000755                  TYPE,SLASH              ;ECHO SLASH
2218  004544  000241                          CLC                     ;CLEAR CARRY
2219  004546  006015                          ROR     (R5)            ;SHIFT LAST TYPED CHARACTER
2220  004550  006215                          ASR     (R5)            ;OUT OF DATA WORD
2221  004552  006215                          ASR     (R5)
2222  004554  000757                          BR      1$              ;GO WAIT FOR NEXT CHARACTER
2223
2224  004556  122700  000015          2$:     CMPB    #15,R0          ;CHECK IF CARRIAGE RETURN
2225  004562  001004                          BNE     3$              ;BRANCH IF NOT CARRIAGE RETURN
2226  004564  000004  000752                  TYPE,CRLF
2227  004570  005725                          TST     (R5)+           ;STEP RETURN ADDRESS
2228  004572  000205                          RTS     R5              ;RETURN
2229
2230  004574  110067  174140          3$:     MOVB    R0,ECHO
2231  004600  000004  000740                  TYPE,ECHO
2232  004604  042700  177770                  BIC     #177770,R0      ;STRIP NON-ESSENTIAL BITS
2233  004610  006315                          ASL     (R5)            ;SHIFT LAST CHARACTER 3 PLACES
2234  004612  006315                          ASL     (R5)            ;LEFT
2235  004614  006315                          ASL     (R5)
2236  004616  050015                          BIS     R0,(R5)         ;AND INSERT NEW CHARACTER
2237  004620  000735                          BR      1$              ;WAIT FOR NEXT CHARACTER
2238
```

```
                                                .SBTTL  PARITY ERROR SERVICE
                                        ;PARITY ERROR SERVICE ROUTINE
004622  005737  177570          .PARSRV:TST     @#SWR           ;CHECK IF HALT ON ERROR
004626  100001                  BPL     1$              ;BRANCH IF NOT HALT ON ERROR
004630  000000                  HALT
004632  000004  000707          1$:     TYPE,PARERRR
004636  110637  000761          MOVB    SP,@#PEFLG      ;SET PARITY ERROR INDICATOR
004642  000137  003416          JMP     @#.HLT          ;GO TO ERROR SERVICE
004646  105037  000761  PERET:  CLRB    @#PEFLG         ;CLEAR PARITY ERROR FLAG
004652  005001                  CLR     R1
004654  005737  000766          TST     @#OPT.CP        ;CHECK IF MEM MGMT IS AVAIL
004660  100032                  BPL     1$              ;BRANCH IF NOT AVAILABLE
004662  012702  077406          MOV     #77406,R2       ;SET UP MEM MGMT
004666  005037  172340          CLR     @#KIPAR0
004672  010237  172300          MOV     R2,@#KIPDR0
004676  012737  000200  172342  MOV     #200,@#KIPAR1
004704  010237  172302          MOV     R2,@#KIPDR1
004710  012737  000400  172344  MOV     #400,@#KIPAR2
004716  010237  172304          MOV     R2,@#KIPDR2
004722  005037  172306          CLR     @#KIPDR3
004726  012737  007600  172356  MOV     #7600,@#KIPAR7
004734  010237  172316          MOV     R2,@#KIPDR7
004740  012737  000001  177572  MOV     #1,@#SR0        ;ENABLE MEM MGMT
004746  012737  004774  000114  1$:     MOV     #2$,@#PARVEC    ;SET NEW PARITY ERROR TRAP VECTOR
004754  012737  005154  000004          MOV     #7$,@#ERRVEC    ;SET TIME OUT TRAP
004762  012737  005166  000250          MOV     #8$,@#MMVEC     ;SET MEM MGMT ABORT VECTOR

004770  005721                  TST     (R1)+           ;SCAN MEMORY FOR PARITY ERROR
004772  000776                  BR      .-2

004774  000004  005210          2$:     TYPE,ADRSIS
005000  004767  176276                  JSR     PC,TYPADR               ;TYPE ADDRESS
005004  000005                  RESET                   ;DISABLE PARITY ERROR DETECTION & MEM MGMT
005006  005737  000766          TST     @#OPT.CP        ;BRANCH IF MEM MGMT NOT AVAILABLE
005012  100002                  BPL     3$
005014  005237  177572          INC     @#SR0           ;RE-ENABLE MEM MGMT
005020  005002                  3$:     CLR     R2              ;INITIALIZE DATA FOR DATA SCAN
005022  014103                  MOV     -(R1),R3        ;GET DATA IN FAILING ADDRESS
005024  010211                  4$:     MOV     R2,(R1)         ;LOAD BINARY COUNT INTO ADDRESS
005026  021102                  CMP     (R1),R2         ;BRANCH IF DATA DOES NOT COMPARE
005030  001015                  BNE     5$
005032  005102                  COM     R2              ;COMPLEMENT DATA
005034  010211                  MOV     R2,(R1)         ;LOAD COMPLEMENT DATA INTO FAILING ADDRESS
005036  021102                  CMP     (R1),R2         ;BRANCH IF DATA DOES NOT COMPARE
005040  001012                  BNE     5$
005042  005402                  NEG     R2              ;STEP DATA
005044  001367                  BNE     4$
005046  000004  005235          TYPE,NOTFND             ;TYPE PARITY ERROR NOT FOUND ON
005052  000004  005312          TYPE,DSCAN              ;DATA SCAN ORIG DATA =
005056  010302                  MOV     R3,R2           ;GET ORIGINAL DATA
005060  004767  176204          JSR     PC,TYPDAT       ;TYPE ORIGINAL DATA
005064  000411                  BR      6$              ;EXIT VIA 6$

005066  000004  005341          5$:     TYPE,GDDAT              ;TYPE GOOD DATA =
005072  004767  176172          JSR     PC,TYPDAT       ;AND THE GOOD DATA
005076  000004  005354          TYPE,BDDAT              ;TYPE BAD DATA =
```

```
2295  005102  011102                          MOV     (R1),R2         ;GET BAD DATA
2296  005104  004767  176160                  JSR     PC,TYPDAT       ;TYPE BAD DATA
2297
2298  005110  000004  000752          6$:     TYPE,CRLF
2299  005114  005737  177570                  TST     @#SWR                   ;CHECK FOR HALT ON ERROR
2300  005120  100001                          BPL     .+4
2301  005122  000000                          HALT
2302  005124  000005                          RESET                           ;DISABLE MEM MGMT & PARITY
2303  005126  012737  004622  000114          MOV     #.PARSRV,@#PARVEC       ;RESET PARITY ERROR TRAP
2304  005134  012737  005540  000004          MOV     #ERPRT,@#ERRVEC         ;AND ERROR VECTOR
2305  005142  012737  000252  000250          MOV     #MMVEC+2,@#MMVEC        ;RESET MEM MGMT ABORT TRAP
2306  005150  000137  006050                  JMP     @#START3                ;RESTART TEST
2307
2308  005154  000004  005235          7$:     TYPE,NOTFND
2309  005160  000004  005275                  TYPE,ASCAN
2310  005164  000751                          BR      6$
2311
2312                                  ;MEMORY MANAGEMENT ABORT ROUTINE
2313  005166  062737  000200  172344  8$:     ADD     #200,@#KIPAR2           ;ADJUST PHYSICAL ADDRESS
2314  005174  012701  020000                  MOV     #20000,R1               ;RESET VIRTUAL ADDRESS
2315  005200  012737  000001  177572          MOV     #1,@#SR0                ;RESET ERROR AND ENABLE
2316  005206  000002                          RTI                             ;RETURN
2317
2318  005210  005015  042515  047515  ADRSIS: .ASCIZ  <15><12>'MEMORY ADDRESS IS '
2319  005216  054522  040440  042104
2320  005224  042522  051523  044440
2321  005232  020123     000
2322  005235     015  050012  051101  NOTFND: .ASCIZ  <15><12>'PARITY ERROR NOT DETECTED ON '
2323  005242  052111  020131  051105
2324  005250  047522  020122  047516
2325  005256  020124  042504  042524
2326  005264  052103  042105  047440
2327  005272  020116     000
2328  005275     101  042104  042522  ASCAN:  .ASCIZ  'ADDRESS SCAN'
2329  005302  051523  051440  040503
2330  005310  000116
2331  005312  040504  040524  051440  DSCAN:  .ASCIZ  'DATA SCAN ORIG DATA = '
2332  005320  040503  020116  051117
2333  005326  043511  042040  052101
2334  005334  020101  020075     000
2335  005341     040  042107  042040  GDDAT:  .ASCIZ  ' GD DATA= '
2336  005346  052101  036501  000040
2337  005354  041040  020104  040504  BDDAT:  .ASCIZ  ' BD DATA= '
2338  005362  040524  020075     000
2339          005370                          .EVEN
```

```
                                                .SBTTL  MISC SUBROUTINES
                                        ;ROUTINE TO SAVE REGISTERS ON THE STACK
                                        ;CALLED BY SAVE MACRO OR JSR    PC,$SAVR
005370  010546                          $SAVR:  MOV     %5,-(SP)
005372  010446                                  MOV     %4,-(SP)
005374  010346                                  MOV     %3,-(SP)
005376  010246                                  MOV     %2,-(SP)
005400  010146                                  MOV     %1,-(SP)
005402  010046                                  MOV     %0,-(SP)
005404  016607  000014                          MOV     14(SP),PC               ;RETURN

                                        ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
                                        ;CALLED BY RESTORE MACRO OR JSR  PC,$RESTR
005410  012666  000014                  $RESTR: MOV     (SP)+,14(SP)    ;SAVE RETURN PC
005414  012600                                  MOV     (SP)+,%0
005416  012601                                  MOV     (SP)+,%1
005420  012602                                  MOV     (SP)+,%2
005422  012603                                  MOV     (SP)+,%3
005424  012604                                  MOV     (SP)+,%4
005426  012605                                  MOV     (SP)+,%5
005430  000207                                  RTS     PC

                                        ;SUBROUTINE TO LOAD DISPLAY REGISTER
005432  013727  001000                  LDDISP: MOV     @#ICNT,(PC)+            ;LOAD PASSCOUNT
005436  000000                          DISPLY: .WORD   0
005440  012746                                  MOV     (PC)+,-(SP)             ;GET SECTION #
005442  000000                          SECT:   .WORD   0
005444  006316                                  ASL     (SP)
005446  006316                                  ASL     (SP)
005450  006316                                  ASL     (SP)
005452  052667  177760                          BIS     (SP)+,DISPLY            ;LOAD SECTION #
005456  113767  001011  177753                  MOVB    @#FRSTAD+1,DISPLY+1     ;LOAD BASE ADDRESS
005464  105737  000770                          TSTB    @#MMON                  ;CHECK IF MEM MGMT IS ON
005470  001403                                  BEQ     1$                      ;BRANCH IF OFF
005472  013737  172344  005436                  MOV     @#KIPAR2,@#DISPLY       ;LOAD CONTENTS OF KIPAR2
005500  013737  005436  177570          1$:     MOV     @#DISPLY,@#DISPLAY      ;DISPLAY IN DISPLAY REGISTER
005506  000207                                  RTS     PC                      ;RETURN
```

```
                                                .SBTTL  KT ABORT, RESERVED & ERROR TRAP SERVICE
                                        ;MEMORY MANAGEMENT ABORT SERVICE ROUTINE
005510  012737  005623  005572  KTABRT: MOV     #KTAMSG,@#ERTAG ;SET UP KT11 ABORT MSG
005516  013716  177576                  MOV     @#SR2,(SP)      ;PUT SR2 ONTO STACK
005522  062716  000002                  ADD     #2,(SP)
005526  000416                          BR      ERRPRT

                                        ;RESERVED INSTRUCTION TRAP SERVICE ROUTINE
005530  012737  005640  005572  RESERR: MOV     #RESMSG,@#ERTAG ;LOAD RESERVED TRAP MESSAGE
005536  000412                          BR      ERRPRT

                                        ;TRAP TO 4 ERROR SERVICE ROUTINE
005540  012737  000340  177776  ERPRT:  MOV     #PRTY7,@#PSW    ;SET PRIORITY LEVEL 7
005546  005737  005566                  TST     @#ERFLAG        ;CHECK IF LAST ERROR TRAP HAS BEEN
005552  001401                          BEQ     .+4             ;REPORTED
005554  000000                          HALT                    ;ERROR! TRAPPING TO LOCATION 4
                                        ;STACK CONTENTS:
                                        ;       (SP)            ;THIS TRAP PC
                                        ;       2(SP)           ;THIS TRAP PSW
                                        ;       4(SP)           ;FIRST TRAP PC
                                        ;       6(SP)           ;FIRST TRAP PSW

005556  012737  005604  005572          MOV     #ERMSG,@#ERTAG  ;SET UP TIME OUT TRAP MSG
005564  005227                  ERRPRT: INC     (PC)+
005566  000000                  ERFLAG: .WORD   0
005570  000004                          TYPE
005572  000000                  ERTAG:  .WORD   0               ;CONTAINS ADR OF ERROR MSG
005574  005037  000772                  CLR     @#DEVID                 ;SET DEVICE ID = CP
005600  000137  003416                  JMP     @#.HLT

005604  005015  051124  050101  ERMSG:  .ASCIZ  <15><12> 'TRAPPED TO 4'
005612  042520  020104  047524
005620  032040    000
005623    015   045412  030524  KTAMSG: .ASCIZ  <15><12> 'KT11 ABORT'
005630  020061  041101  051117
005636  000124
005640  005015  042522  042523  RESMSG: .ASCIZ  <15><12>'RESERVED INST TRAP'
005646  053122  042105  044440
005654  051516  020124  051124
005662  050101    000

        005666                          .EVEN

                                        .SBTTL  PROGRAM INITIALIZATION
005666  000005                  START:  RESET
005670  012706  000600                  MOV     #KPTR,SP        ;SET KERNEL STACK PTR

                                        ;DETERMINE IF PROGRAM LOADED VIA ACT11 IN QUICK VERIFY MODE
005674  105037  000771                  CLRB    @#QV                    ;SET IND NOT QV MODE
005700  005737  000042                  TST     @#42                    ;BRANCH IF NOT VIA ACT11
005704  001405                          BEQ     1$
005706  005737  032622                  TST     @#LOGICAL+4             ;BANCH IF NOT QV
005712  100002                          BPL     1$
005714  110637  000771                  MOVB    SP,@#QV                 ;SET ACT11 QV MODE
                                        ;ROUTINE TO DETERMINE LAST MEMORY ADDRESS
005720  012737  005742  000004  1$:     MOV     #2$,@#ERRVEC            ;SET TIME OUT TRAP TO RETURN
005726  012737  000002  000006          MOV     #RTI,@#ERRVEC+2
```

```
2434  005734  005000                          CLR     R0
2435  005736  005720                          TST     (R0)+                   ;WILL TIME OUTWHEN END OF MEMORY
2436  005740  000776                          BR      .-2
2437  005742  162700  000002          2$:     SUB     #2,R0
2438  005746  010027                          MOV     R0,(PC)+                ;SET VALUE INTO LSTMEM
2439  005750  000000                LSTMEM:   .WORD   0                       ;CONTAINS VALUE OF LAST MEMORY ADDRESS
2440  005752  105737  000771                  TSTB    @#QV                    ;NO NEED TO PRESERVE LOADERS
2441  005756  001003                          BNE     1$                      ;IF QV
2442  005760  162737  004000  005750          SUB     #4000,@#LSTMEM          ;SET PROTECTION FOR LOADERS
2443  005766  012737  033012  001012  1$:     MOV     #ENDTAG+2,@#FRSTMEM     ;SET LOWER BOUNDARY
2444  005774  000425                          BR      START3                  ;GO TO START3
2445
2446                                          ;PROGRAM STARTS HERE WHEN ADDRESS 204 IS USED AS STARTING ADDRESS.
2447  005776  012706  000600          START1: MOV     #KPTR,SP                ;SET STACK PTR
2448  006002  012737  002736  000020          MOV     #.TYPE,@#IOTVEC         ;SET IOT VECTOR TO TYPE ROUTINE
2449  006010  000004  032710                  TYPE,MSG1
2450  006014  004567  176470                  JSR     R5,RECO                 ;GET LOWER LIMIT
2451  006020  000000                  1$:     .WORD   0                       ;CONTAINS TYPED LOWER LIMIT
2452  006022  016737  177772  001012          MOV     1$,@#FRSTMEM            ;SET IN LOWER LIMIT
2453  006030  000004  032725                  TYPE,MSG2
2454  006034  004567  176450                  JSR     R5,RECO                 ;GET UPPER LIMIT
2455  006040  000000                  2$:     .WORD   0                       ;CONTAINS UPPER LIMIT
2456  006042  016737  177772  005750          MOV     2$,@#LSTMEM
2457
2458                                          ;PROGRAM STARTS HERE WHEN ADDRESS 210 IS USED AS STARTING ADDRESS.
2459  006050  012706  000600          START3: MOV     #KPTR,SP                ;SET STACK PTR
2460  006054  005037  001000                  CLR     @#ICNT                  ;CLEAR PASS COUNT
2461  006060  105037  000770                  CLRB    @#MMON                  ;SET MEM MGMT ON IND=NOT ON
2462  006064  004737  000120                  JSR     PC,@#.MAMF              ;GO ENABLE PARITY IF AVAILABLE
2463  006070  012737  001600  032432          MOV     #1600,@#NEXPAR
2464  006076  012737  020040  001150          MOV     #20040,@#ITCNT          ;SET TEST ITERATION SOUNT
2465  006104  105737  000771                  TSTB    @#QV                    ;BRANCH IF NOT IN QV MODE
2466  006110  001403                          BEQ     START2
2467  006112  012737  000401  001150          MOV     #401,@#ITCNT            ;SET 1 ITERATION FOR TESTS
2468
2469
2470                                          ;PROGRAM RESTARTS HERE AFTER RELOCATION ABOVE 28K IS COMPLETE.
2471  006120  012706  000500          START2: MOV     #STKPTR,SP              ;SET STACK PTR
2472  006124  012737  005540  000004          MOV     #ERPRT,@#ERRVEC         ;SET ERROR TRAP
2473  006132  012737  005530  000010          MOV     #RESERR,@#RESVEC        ;SET RESERVED INST TRAP VECTOR
2474  006140  012737  000002  000012          MOV     #RTI,@#RESVEC+2
2475  006146  012737  000610  000024          MOV     #PDWN,@#PFVEC           ;SET POWER FAIL TRAP VECTOR
2476  006154  012737  000340  000026          MOV     #340,@#PFVEC+2          ;AND PRIORITY LEVEL 7
2477  006162  012737  005510  000250          MOV     #KTABRT,@#MMVEC         ;SET KT11 ABORT VECTOR
2478  006170  012737  002736  000020          MOV     #.TYPE,@#IOTVEC         ;SET IOT VECTOR TO TYPE ROUTINE
2479  006176  012737  000340  000022          MOV     #PRTY7,@#IOTVEC+2       ;SET LEVEL 7 ON TRAP
2480  006204  012737  001014  000030          MOV     #SCOPEA,@#EMTVEC        ;SET EMT(SCOPE) TRAP VECTOR
2481  006212  012737  003416  000034          MOV     #.HLT,@#TRAPVEC         ;SET TRAP (HLT) VECTOR
2482  006220  012737  000340  000036          MOV     #340,@#TRAPVEC+2        ;PRIORITY LEVEL 7 ON TRAP
2483  006226  005037  005566                  CLR     @#ERFLAG                ;CLEAR ABORT & TRAP TO 4 FLAG
2484  006232  005037  000772                  CLR     @#DEVID
2485  006236  004737  005432                  JSR     PC,@#LDDISP             ;LOAD DISPLAY REGISTER
2486  006242  105037  000761                  CLRB    @#PEFLG                 ;CLEAR PARITY ERROR FLAG
2487  006246  052737  000100  177560          BIS     #100,@#TKS              ;SET IE BIT IN KEYBOARD STATUS REG
2488  006254  005027                          CLR     (PC)+                   ;CLEAR 'NO TYPING' INDICATOR
2489  006256  000000                NOTYPE:   .WORD   0
```

```
2490
2491                                    ;THE BELOW ROUTINE ASCERTAINS WHICH CP & CP OPTIONS THE PROGRAM IS RUN-
2492                                    ;NING ON AND SETS AN INDICATOR IN OPT.CP ACCORDINGLY.
2493  006260  012737  000006  000004  CPCHK:  MOV     #ERRVEC+2,@#ERRVEC      ;SET UP ERROR TRAP TO RETURN
2494  006266  012737  000012  000010          MOV     #RESVEC+2,@#RESVEC      ;AND ALSO RESERVED INST TRAP
2495  006274  012700  000004                  MOV     #4,R0
2496  006300  000261                          SEC
2497  006302  005037  177766                  CLR     @#CPUERR                ;CLEAR CPU ERROR FLG
2498  006306  005600                          SBC     R0
2499  006310  000261                          SEC
2500  006312  005737  177772                  TST     @#PIRQ                  ;R0=3 IF 11/45
2501  006316  005600                          SBC     R0                      ;R0=2 IF 11/40
2502  006320  000261                          SEC
2503  006322  105737  177777                  TSTB    @#PSW+1                 ;TIMES OUT IF 11/20
2504  006326  005600                          SBC     R0                      ;R0=1 IF 11/20
2505  006330  005037  177700                  CLR     @#177700                ;CLEARS R0 IF 11/05
2506  006334  006300                          ASL     R0                      ;SHIFT CP INDICATOR
2507  006336  010002                          MOV     R0,R2                   ;MOVE CP TYPE TO R2
2508  006340  000261                          SEC
2509  006342  005737  177572                  TST     @#SR0                   ;CHECK IF MEM MGMT IS AVAILABLE
2510  006346  103402                          BCS     1$
2511  006350  052702  100000                  BIS     #KTOPT,R2               ;SET MEM MGMT AVAIL INDICATOR
2512  006354  005004                  1$:     CLR     R4
2513  006356  000261                          SEC
2514  006360  072404                          ASH     R4,R4                   ;WILL TRAP IF 11/40 WITHOUT EIS
2515  006362  103402                          BCS     2$                      ;BRANCH IF NO EIS AVAILABLE
2516  006364  052702  040000                  BIS     #EISOPT,R2              ;SET EIS AVAIL INDICATOR
2517  006370  000261                  2$:     SEC                             ;SET CARRY
2518  006372  170500                          TSTF    R0                      ;WILL CLEAR CARRY IF 11/45 FLOATING POINT
2519  006374  170000                          CFCC                            ;IS AVAIL. COPY FLOATING CC'S INTO PSW
2520  006376  103402                          BCS     3$                      ;BRANCH IF NO FLOATING POINT
2521  006400  052702  020000                  BIS     #FPOPT,R2               ;SET FP OPTION AVAIL INDICATOR
2522  006404  000261                  3$:     SEC                             ;SET 'C' BIT
2523  006406  075000                          FADD    R0
2524  006410  103402                          BCS     4$                      ;BRANCH IF NO FIS OPTION
2525  006412  052702  010000                  BIS     #FISOPT,R2              ;SET FIS OPTION AVAIL INDICATOR
2526  006416  000261                  4$:     SEC                             ;SET 'C' BIT
2527  006420  005037  177774                  CLR     @#SLR                   ;CLEAR STACK LIMIT REGISTER
2528  006424  103402                          BCS     5$                      ;BRANCH IF NOT AVAILABLE
2529  006426  052702  004000                  BIS     #KJOPT,R2               ;SET KJ OPTION AVAIL INDICATOR
2530  006432  000261                  5$:     SEC
2531  006434  005737  172540                  TST     @#PLKCSR                ;BRANCH IF NO KW11-P
2532  006440  103402                          BCS     6$
2533  006442  052702  002000                  BIS     #PLKOPT,R2              ;SET OPTION INDICATOR
2534  006446  000261                  6$:     SEC
2535  006450  005737  177546                  TST     @#LKS                   ;BRANCH IF NO KW11-L
2536  006454  103402                          BCS     7$
2537  006456  052702  001000                  BIS     #LKOPT,R2               ;SET OPTION INDICATOR
2538  006462  000261                  7$:     SEC
2539  006464  005737  177564                  TST     @#TPS                   ;BRANCH IF NO CONSOLE TTY
2540  006470  103402                          BCS     8$
2541  006472  052702  000400                  BIS     #TTOPT,R2
2542  006476  012737  005540  000004  8$:     MOV     #ERPRT,@#ERRVEC         ;RESTORE ERROR TRAP
2543  006504  012737  005530  000010          MOV     #RESERR,@#RESVEC        ;AND ALSO RESERVED INST TRAP
2544  006512  010237  000766                  MOV     R2,@#OPT.CP             ;LOAD INDICATOR
2545  006516  005727                          TST     (PC)+                   ;BRANCH IF OPT.CP HAS BEEN TYPED
```

```
2546  006520  000000                  9$:     .WORD   0
2547  006522  001010                          BNE     DEVCHK
2548  006524  000004  033012                  TYPE,AOPT.CP
2549  006530  004767  174534                  JSR     PC,TYPDAT
2550  006534  000004  000752                  TYPE,CRLF
2551  006540  005267  177754                  INC     9$                      ;SET OPT.CP HAS BEEN TYPED IND.
2552                                  ;ROUTINE TO DETERMINE WHICH DEVICES ARE ON SYSTEM.
2553  006544  012737  000006  000004  DEVCHK: MOV     #ERRVEC+2,@#ERRVEC      ;SET UP TIME OUT VECTOR
2554  006552  012705  002626                  MOV     #VALDEV,R5              ;GET STARTING ADDRESS OF TABLE
2555  006556  005002                          CLR     R2
2556  006560  005205                  DEVSTA: INC     R5                      ;MOVE POINTER UP ONE
2557  006562  020567  174060                  CMP     R5,VALDEV+20            ;ARE WE DONE
2558  006566  001017                          BNE     2$                      ;NO GO ON
2559  006570  012737  005540  000004          MOV     #ERPRT,@#ERRVEC         ;IF SO RESTORE TIME OUT VECTOR
2560  006576  005727                          TST     (PC)+                   ;TEST IF DEVOPT HES BEEN PRINTED
2561  006600  000000                  1$:     .WORD   0
2562  006602  001010                          BNE     3$
2563  006604  000004  033066                  TYPE,OPTDEV
2564  006610  004767  174454                  JSR     PC,TYPDAT
2565  006614  000004  000752                  TYPE,CRLF
2566  006620  005267  177754                  INC     1$
2567  006624  000502                  3$:     BR      ENDSIZ                  ;AND BRANCH
2568  006626  000261                  2$:     SEC                             ;SET 'C' BIT FOR TRAP INDICATOR
2569  006630  105715                          TSTB    (R5)                    ;TEST IF DEVICE PRESENT
2570  006632  103474                          BCS     NOTHOM                  ;IF NOT PRESENT GO TO ROUTINE
2571                                                                          ;TO CLR LOCATION IN TABLE
2572  006634  020567  173770                  CMP     R5,VALDEV+2             ;THIS PORTION OF
2573  006640  001422                          BEQ     RK                      ;CODE DETERMINES
2574  006642  020567  173764                  CMP     R5,VALDEV+4             ;WHICH DEVICES
2575  006646  001422                          BEQ     RF                      ;DEVICES HAVE
2576  006650  020567  173760                  CMP     R5,VALDEV+6             ;ANSWERED AND
2577  006654  001426                          BEQ     RP                      ;DISPATCHES THE
2578  006656  020567  173754                  CMP     R5,VALDEV+10            ;PROGRAM TO DETERMINE
2579  006662  001432                          BEQ     RC                      ;IF IT WAS REALLY
2580  006664  020567  173750                  CMP     R5,VALDEV+12            ;THAT DEVICE OR
2581  006670  001436                          BEQ     RP4                     ;ANOTHER DEVICE
2582  006672  020567  173744                  CMP     R5,VALDEV+14            ;WITH THE SAME
2583  006676  001441                          BEQ     R5                      ;ADDRESS.
2584  006700  020567  173740                  CMP     R5,VALDEV+16
2585  006704  001441                          BEQ     RK6
2586
2587  006706  052702  000001          RK:     BIS     #RKOPT,R2
2588  006712  000722                          BR      DEVSTA
2589
2590  006714  000261                  RF:     SEC                             ;SET TIME OUT VECTOR
2591  006716  005767  170526                  TST     177450                  ;IF 'C'BIT SET TRAPPED
2592  006722  103040                          BCC     NOTHOM                  ;MUST BE RF11
2593  006724  052702  000002                  BIS     #RFOPT,R2
2594  006730  000713                          BR      DEVSTA
2595
2596  006732  033767  176726  011040  RP:     BIT     @#176726,20000          ;LOOKING AT DEVICE TYPE
2597  006740  001031                          BNE     NOTHOM                  ;REGISTER IF NO MATCH
2598  006742  052702  000004                  BIS     #RPOPT,R2
2599  006746  000704                          BR      DEVSTA                  ;MUST BE RP11
2600
2601  006750  000261                  RC:     SEC                             ;SET TIME OUT VECTOR
```

```
2602  006752  005767  170512          TST     177470          ;IF 'C'BIT SET TRAPPED
2603  006756  103022                  BCC     NOTHOM          ;MUST BE RC11
2604  006760  052702  000010          BIS     #RCOPT,R2
2605  006764  000675                  BR      DEVSTA
2606
2607  006766  005767  173642  RP4:    TST     VALDEV+6        ;SOMETHING ANSWERED
2608  006772  001014                  BNE     NOTHOM          ;IF NOT RP11 MUST
2609  006774  052702  000020          BIS     #RP4OPT,R2
2610  007000  000667                  BR      DEVSTA          ;BE RP04/05/06
2611
2612  007002  052702  000040  RS:     BIS     #RSOPT,R2
2613  007006  000664                  BR      DEVSTA
2614
2615  007010  005767  173622  RK6:    TST     VALDEV+10       ;SOMETHING ANSWERED
2616  007014  001003                  BNE     NOTHOM          ;IF NOT RC11 MUST
2617  007016  052702  000100          BIS     #RK6OPT,R2
2618  007022  000656                  BR      DEVSTA          ;BE RK06
2619
2620  007024  005015  NOTHOM: CLR     (R5)
2621  007026  000167  177526          JMP     DEVSTA
2622
2623  007032  ENDSIZ:
2624
```

```
                                                .SBTTL  START OF SECTION 0
                                        ;000000000000 FIRST ADDRESS TO BE RELOCATED 000000000
007032  010700                          RELO:   MOV     PC,R0           ;GET PC
007034  005740                                  TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF RELO
007036  010037  001010                          MOV     R0,@#FRSTAD     ;SAVE
007042  012737  000000  005442                  MOV     #0,@#SECT       ;SET SECTION #
007050  004737  005432                          JSR     PC,@#LDDISP     ;LOAD DISPLAY REG
007054  013767  005436  000746                  MOV     @#DISPLY,RELOO
007062  010700                                  MOV     PC,R0           ;GET CURRENT PC
007064  162700  007064                          SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
007070  010037  001004                          MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
007074  010701                                  MOV     PC,R1           ;SET NEW SCOPE PTR
                                        ;MEMORY AND DISK (IF SELECTED) VERIFICATION TEST.
007076  000167  000714                          JMP     1$
007102  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007110  177777  000000  000000
007116  000000  000000
007122  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007130  177777  000000  000000
007136  000000  000000
007142  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007150  177777  000000  000000
007156  000000  000000
007162  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007170  177777  000000  000000
007176  000000  000000
007202  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007210  177777  000000  000000
007216  000000  000000
007222  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007230  177777  000000  000000
007236  000000  000000
007242  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007250  177777  000000  000000
007256  000000  000000
007262  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007270  177777  000000  000000
007276  000000  000000
007302  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007310  177777  000000  000000
007316  000000  000000
007322  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007330  177777  000000  000000
007336  000000  000000
007342  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007350  177777  000000  000000
007356  000000  000000
007362  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007370  177777  000000  000000
007376  000000  000000
007402  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007410  177777  000000  000000
007416  000000  000000
007422  177777  177777  177777                  .WORD   -1,-1,-1,-1,0,0,0,0
007430  177777  000000  000000
```

```
2681  007436  000000  000000
2682  007442  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2683  007450  177777  000000  000000
2684  007456  000000  000000
2685  007462  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2686  007470  177777  000000  000000
2687  007476  000000  000000
2688  007502  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2689  007510  177777  000000  000000
2690  007516  000000  000000
2691  007522  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2692  007530  177777  000000  000000
2693  007536  000000  000000
2694  007542  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2695  007550  177777  000000  000000
2696  007556  000000  000000
2697  007562  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2698  007570  177777  000000  000000
2699  007576  000000  000000
2700  007602  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2701  007610  177777  000000  000000
2702  007616  000000  000000
2703  007622  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2704  007630  177777  000000  000000
2705  007636  000000  000000
2706  007642  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2707  007650  177777  000000  000000
2708  007656  000000  000000
2709  007662  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2710  007670  177777  000000  000000
2711  007676  000000  000000
2712  007702  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2713  007710  177777  000000  000000
2714  007716  000000  000000
2715  007722  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2716  007730  177777  000000  000000
2717  007736  000000  000000
2718  007742  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2719  007750  177777  000000  000000
2720  007756  000000  000000
2721  007762  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0,0,0
2722  007770  177777  000000  000000
2723  007776  000000  000000
2724  010002  177777  177777  177777          .WORD   -1,-1,-1,-1,0,0
2725  010010  177777  000000  000000
2726  010016                          1$:
2727  010016  010702                          MOV     PC,R2
2728  010020  062702  000012                  ADD     #12,R2
2729  010024  012707  001152                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
2730  010030  000000                  RELOO:  .WORD   0
2731                                  ;0000000000000 LAST ADDRESS OF CODE TO BE RELOCATED 00000000000
2732
2733
2734                                          .SBTTL  START OF SECTION 1
2735                                  ;1111111111111 FIRST ADDRESS TO BE RELOCATED 111111111
2736  010032  010700                  REL1:   MOV     PC,R0           ;GET PC
```

```
2737  010034  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL1
2738  010036  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
2739  010042  012737  000001  005442          MOV     #1,@#SECT       ;SET SECTION #
2740  010050  004737  005432                  JSR     PC,@#LDDISP     ;LOAD DISPLAY GEG
2741  010054  013767  005436  003776          MOV     @#DISPLY,REL11
2742  010062  010700                          MOV     PC,R0           ;GET CURRENT PC
2743  010064  162700  010064                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
2744  010070  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
2745  010074  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR
2746                                  ;CHECK BRANCH INSTRUCTIONS
2747  010076  000257                          CCC                     ;CC'S=0000
2748  010100  103407                          BCS     CC0             ;SAME AS BLO
2749  010102  102406                          BVS     CC0
2750  010104  001405                          BEQ     CC0
2751  010106  100404                          BMI     CC0
2752  010110  002403                          BLT     CC0
2753  010112  003402                          BLE     CC0
2754  010114  101401                          BLOS    CC0
2755  010116  101001                          BHI     .+4
2756  010120  104400                  CC0:    HLT                     ;ONE OF THE ABOVE BRANCHES FAILED
2757
2758                                  ;CONTINUE
2759  010122  000270                          SEN                     ;CC'S=1000
2760  010124  100003                          BPL     CC1
2761  010126  002002                          BGE     CC1
2762  010130  003001                          BGT     CC1
2763  010132  002401                          BLT     .+4
2764  010134  104400                  CC1:    HLT                     ;ONE OF THE ABOVE BRANCHES FAILED
2765
2766                                  ;CONTINUE
2767  010136  000262                          SEV                     ;CC'S=1010
2768  010140  102003                          BVC     CC2
2769  010142  002402                          BLT     CC2
2770  010144  003401                          BLE     CC2
2771  010146  002001                          BGE     .+4
2772  010150  104400                  CC2:    HLT                     ;ERROR! ONE OF THE ABOVE BRANCHES FAILED
2773
2774                                  ;CONTINUE
2775  010152  000261                          SEC                     ;CC'S=1011
2776  010154  103002                          BCC     CC3
2777  010156  101001                          BHI     CC3
2778  010160  003001                          BGT     .+4
2779  010162  104400                  CC3:    HLT                     ;ERROR! ONE OF THE ABOVE BRANCHES FAILED
2780
2781                                  ;CONTINUE
2782  010164  000264                          SEZ                     ;CC'S=1111
2783  010166  001003                          BNE     CC4
2784  010170  003002                          BGT     CC4
2785  010172  101001                          BHI     CC4
2786  010174  003401                          BLE     .+4
2787  010176  104400                  CC4:    HLT                     ;ERROR! ONE OF THE ABOVE BRANCHES FAILED
2788  010200  104000                          SCOPE
2789
2790                                  ;TEST UNARY CONDITION CODES
2791                                  ;CLR    R0
2792  010202  000277                          SCC
```

```
2793  010204  000244                    CLZ
2794  010206  005000                    CLR     R0              ;R0=0,CC'S=0100
2795  010210  103404                    BCS     CLRO
2796  010212  102403                    BVS     CLRO
2797  010214  001002                    BNE     CLRO
2798  010216  100401                    BMI     CLRO
2799  010220  003401                    BLE     .+4
2800  010222  104400            CLRO:   HLT                     ;ERROR! INCORRECT CC'S AFTER CLR
2801
2802  010224  000277                    SCC
2803  010226  000244                    CLZ
2804  010230  005700                    TST     R0              ;R0=0,CC'S=0100
2805  010232  103404                    BCS     TSTO
2806  010234  102403                    BVS     TSTO
2807  010236  001002                    BNE     TSTO
2808  010240  100401                    BMI     TSTO
2809  010242  101401                    BLOS    .+4
2810  010244  104400            TSTO:   HLT                     ;ERROR! INCORRECT CC'S AFTER TST
2811
2812  010246  000257                    CCC
2813  010250  000266                    +SEZ!SEV
2814  010252  005100                    COM     R0              ;R0=-1,CC'S=1001
2815  010254  103004                    BCC     COMO
2816  010256  102403                    BVS     COMO
2817  010260  001402                    BEQ     COMO
2818  010262  100001                    BPL     COMO
2819  010264  002401                    BLT     .+4
2820  010266  104400            COMO:   HLT                     ;ERROR! INCORRECT CC'S AFTER COM
2821
2822  010270  000261                    SEC
2823  010272  005500                    ADC     R0              ;R0=000000,CC'S=0101
2824  010274  103003                    BCC     ADCO
2825  010276  102402                    BVS     ADCO
2826  010300  001001                    BNE     ADCO
2827  010302  002001                    BGE     .+4
2828  010304  104400            ADCO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ADC
2829
2830  010306  000261                    SEC
2831  010310  006000                    ROR     R0              ;R0=100000,CC'S=1010
2832  010312  103404                    BCS     RORO
2833  010314  102003                    BVC     RORO
2834  010316  001402                    BEQ     RORO
2835  010320  100001                    BPL     RORO
2836  010322  003001                    BGT     .+4
2837  010324  104400            RORO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ROR
2838  010326  000277                    SCC
2839  010330  000242                    CLV
2840  010332  005300                    DEC     R0              ;R0=077777,CC'S=0011
2841  010334  103004                    BCC     DECO
2842  010336  102003                    BVC     DECO
2843  010340  001402                    BEQ     DECO
2844  010342  100401                    BMI     DECO
2845  010344  003401                    BLE     .+4
2846  010346  104400            DECO:   HLT                     ;ERROR! INCORRECT CC'S AFTER DEC
2847
2848  010350  000257                    CCC
```

```
2849  010352  005200                  INC     R0              ;R0=100000,CC'S=1010
2850  010354  103404                  BCS     INCO
2851  010356  102003                  BVC     INCO
2852  010360  001402                  BEQ     INCO
2853  010362  100001                  BPL     INCO
2854  010364  003001                  BGT     .+4
2855  010366  104400          INCO:   HLT                     ;ERROR! INCORRECT CC'S AFTER INC
2856
2857  010370  000277                  SCC
2858  010372  000242                  CLV
2859  010374  005400                  NEG     R0              ;R0=100000,CC'S=1011
2860  010376  103003                  BCC     NEGO
2861  010400  102002                  BVC     NEGO
2862  010402  001401                  BEQ     NEGO
2863  010404  002001                  BGE     .+4
2864  010406  104400          NEGO:   HLT                     ;ERROR! INCORRECT CC'S AFTER NEG
2865
2866  010410  000261                  SEC
2867  010412  006300                  ASL     R0              ;R0=000000,CC'S=0111
2868  010414  103004                  BCC     ASLO
2869  010416  102003                  BVC     ASLO
2870  010420  001002                  BNE     ASLO
2871  010422  100401                  BMI     ASLO
2872  010424  101401                  BLOS    .+4
2873  010426  104400          ASLO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ASL
2874
2875  010430  006100                  ROL     R0              ;R0=000001,CC'S=0000
2876  010432  103402                  BCS     ROLO
2877  010434  003401                  BLE     ROLO
2878  010436  002001                  BGE     .+4
2879  010440  104400          ROLO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ROL
2880
2881  010442  006200                  ASR     R0              ;R0=000000,CC'S=0111
2882  010444  103003                  BCC     ASRO
2883  010446  102002                  BVC     ASRO
2884  010450  001001                  BNE     ASRO
2885  010452  002401                  BLT     .+4
2886  010454  104400          ASRO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ASR
2887
2888  010456  000277                  SCC
2889  010460  005600                  SBC     R0              ;R0=-1,CC'S=1001
2890  010462  103002                  BCC     SBCO
2891  010464  102401                  BVS     SBCO
2892  010466  003401                  BLE     .+4
2893  010470  104400          SBCO:   HLT                     ;ERROR! INCORRECT CC'S AFTER SBC
2894
2895  010472  005400                  NEG     R0              ;R0=000001,CC'S=00001
2896  010474  000300                  SWAB    R0              ;R0=000400',CC'S=0100
2897  010476  103403                  BCS     SWABO
2898  010500  102402                  BVS     SWABO
2899  010502  001001                  BNE     SWABO
2900  010504  002001                  BGE     .+4
2901  010506  104400          SWABO:  HLT                     ;ERROR! INCORRECT CC'S AFTER SWAB
2902  010510  104000                  SCOPE
2903
2904                          ;CHECK REGISTER SELECTION
```

```
2905  010512  005000                        CLR     R0
2906  010514  000277                        SCC
2907  010516  006100                        ROL     R0              ;R0=1
2908  010520  010002                        MOV     R0,R2
2909  010522  006302                        ASL     R2              ;R2=2
2910  010524  010203                        MOV     R2,R3
2911  010526  006303                        ASL     R3              ;R3=4
2912  010530  010304                        MOV     R3,R4
2913  010532  006304                        ASL     R4              ;R4=10
2914  010534  010405                        MOV     R4,R5
2915  010536  006305                        ASL     R5              ;R5=20
2916  010540  010546                        MOV     R5,-(SP)        ;SET BITS SET IN REGISTERS
2917  010542  050416                        BIS     R4,(SP)         ;INTO STACK ADDRESS
2918  010544  050316                        BIS     R3,(SP)
2919  010546  050216                        BIS     R2,(SP)
2920  010550  050016                        BIS     R0,(SP)
2921  010552  022726  000037                CMP     #37,(SP)+
2922  010556  001401                        BEQ     .+4             ;WERE SET
2923  010560  104400                        HLT                     ;MISSING BIT(S) REPRESENT
2924                                                                ;INCORRECT REGISTER SELECTION
2925
2926                                ;CHECK THAT ALL BITS CAN BE SET & CLEARED IN ALL REGISTERS
2927  010562  000257                        CCC
2928  010564  112700  000377                MOVB    #377,R0         ;SET ALL BITS (MOVB EXTENDS SIGN)
2929  010570  006100                1$:     ROL     R0              ;ROTATE A 0 THROUGH ALL BIT
2930  010572  103776                        BCS     1$              ;POSITIONS
2931  010574  005200                        INC     R0              ;FINAL RESULT IS -1
2932  010576  001401                        BEQ     .+4
2933  010600  104400                        HLT                     ;ERROR!
2934
2935  010602  012700  000020                MOV     #16.,R0         ;SET SHIFT COUNT
2936  010606  005002                        CLR     R2
2937  010610  000261                2$:     SEC
2938  010612  006002                        ROR     R2              ;ROTATE 1 THROUGH ALL BIT POSITS
2939  010614  005300                        DEC     R0              ;DECREMENT SHIFT COUNT
2940  010616  001374                        BNE     2$
2941  010620  005102                        COM     R2              ;R2 SHOULD CONTAIN -1
2942  010622  001401                        BEQ     .+4
2943  010624  104400                        HLT                     ;ERROR! CHECK R2 SHOULD = 0
2944
2945  010626  012703  100000                MOV     #100000,R3
2946  010632  006203                3$:     ASR     R3              ;EXTEND 1 BIT THROUGH ALL POSITIONS
2947  010634  103376                        BCC     3$
2948  010636  005203                        INC     R3
2949  010640  001401                        BEQ     .+4
2950  010642  104400                        HLT                     ;ERROR!
2951
2952  010644  112704  177401                MOVB    #177401,R4      ;R4=1
2953  010650  060404                4$:     ADD     R4,R4           ;HAS THE AFFECT OF SHIFTING A BIT
2954  010652  103376                        BCC     4$              ;THROUGH ALL POSITIONS
2955  010654  005704                        TST     R4              ;RESULT SHOULD BE 0
2956  010656  001401                        BEQ     .+4
2957  010660  104400                        HLT
2958
2959  010662  012705  000001                MOV     #1,R5
2960  010666  006305                5$:     ASL     R5
```

```
2961  010670  102376                        BVC     5$
2962  010672  006305                        ASL     R5
2963  010674  103002                        BCC     6$
2964  010676  005705                        TST     R5
2965  010700  001401                        BEQ     .+4
2966  010702  104400                  6$:   HLT
2967
2968                                  ;CHECK REGISTER VOLITILITY
2969  010704  005002                        CLR     R2
2970  010706  005102                        COM     R2                  ;R2=-1
2971  010710  010203                        MOV     R2,R3
2972  010712  000257                        CCC
2973  010714  006002                        ROR     R2                  ;R2=LOOP COUNT
2974  010716  006202                        ASR     R2
2975  010720  010304                  7$:   MOV     R3,R4
2976  010722  005302                        DEC     R2                  ;DECREMENT LOOP COUNT
2977  010724  001375                        BNE     7$
2978  010726  005203                        INC     R3                  ;CHECK R3
2979  010730  001002                        BNE     8$
2980  010732  005204                        INC     R4                  ;CHECK R4
2981  010734  001401                        BEQ     .+4
2982  010736  104400                  8$:   HLT
2983
2984                                  ;CHECK TRANSFER OF REGISTER DATA BETWEEN THE GS AND GD REGISTERS (11/45)
2985  010740  032737  000020  177776  GSTST:  BIT     #20,@#PSW           ;CHECK IF 'T' BIT IS SET
2986  010746  001052                        BNE     7$                  ;SKIP TEST IF 'T' BIT SET
2987  010750  010146                        MOV     R1,-(SP)            ;SAVE SCOPE PTR
2988  010752  010627                        MOV     SP,(PC)+            ;SAVE STACK PTR
2989  010754  000000                  1$:   .WORD   0                   ;CONTAINS SAVED STACK PTR
2990  010756  010727                        MOV     PC,(PC)+            ;LOAD DATA. THE CURRENT PC IS USED AS
2991  010760  000000                  2$:   .WORD   0                   ;DATA. IF THIS TEST FAILS 2$ CON-
2992                                                                    ;TAINS THE DATA BEING USED.
2993  010762  005267  177772               INC     2$                  ;MAKE ODD TO CHECK BIT 0
2994  010766  016700  177766            3$:   MOV     2$,R0               ;LOAD GD REGISTER 0
2995  010772  010001                        MOV     R0,R1               ;TRANSFER GS REG 0 TO GD REG 1
2996  010774  010102                        MOV     R1,R2               ;AND GS REG 1 TO GD REG 2
2997  010776  010203                        MOV     R2,R3               ;ETC...
2998  011000  010304                        MOV     R3,R4
2999  011002  010405                        MOV     R4,R5
3000  011004  152737  000340  177776        BISB    #340,@#PSW          ;SET PRIORITY LEVEL 7
3001  011012  010506                        MOV     R5,SP               ;TRANSFER GS REG 5 TO GD STK PTR
3002  011014  010627                        MOV     SP,(PC)+            ;TRANSFER GS STK PTR TO MEMORY
3003  011016  000000                  4$:   .WORD   0                   ;CONTAINS GS STACK PTR
3004  011020  016706  177730               MOV     1$,SP               ;RESTORE STK PTR NEEDED FOR HLT/SCOPE
3005  011024  142737  000340  177776        BICB    #340,@#PSW          ;SET PRIORITY LEVEL 0
3006  011032  026700  177760               CMP     4$,R0               ;COMPARE GS/GD STKPTR WITH GS REG 0
3007  011036  001004                        BNE     5$                  ;BRANCH IF THEY WERE NOT =
3008  011040  006367  177714               ASL     2$                  ;SHIFT TEST DATA UNTIL = 000000
3009  011044  001350                        BNE     3$
3010  011046  000411                        BR      6$
3011  011050  010046                  5$:   MOV     R0,-(SP)            ;GET GS REG 0
3012  011052  010146                        MOV     R1,-(SP)            ;ETC...
3013  011054  010246                        MOV     R2,-(SP)
3014  011056  010346                        MOV     R3,-(SP)
3015  011060  010446                        MOV     R4,-(SP)
3016  011062  010546                        MOV     R5,-(SP)
```

```
3017  011064  104400                  HLT                     ;ERROR! DATA IN GS STK PTR NOT = GS REG 0
3018                                                          ;GS REG 0-GS REG 5 ARE ON THE STACK
3019  011066  016706  177662          MOV     1$,SP           ;RESTORE STACK PTR
3020  011072  012601            6$:   MOV     (SP)+,R1        ;RESTORE SCOPE PTR
3021  011074  104000            7$:   SCOPE
3022
3023                              ;TEST UNARY WORD INSTRUCTIONS USING ADDRESS MODE 1
3024  011076  000401                  BR      .+4
3025  011100  000000                  .WORD   0               ;RESERVE ADDRESS FOR TESTS
3026  011102  010702                  MOV     PC,R2
3027  011104  162702  000004          SUB     #4,R2           ;R2 POINTS TO RESERVED WORD
3028  011110  005012                  CLR     (R2)            ;PRESET (R2)
3029
3030  011112  000261                  SEC
3031  011114  006012                  ROR     (R2)            ;(R2)=100000,CC=1010
3032  011116  101402                  BLOS    ROR1
3033  011120  100001                  BPL     ROR1
3034  011122  002001                  BGE     .+4
3035  011124  104400            ROR1: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3036
3037  011126  000257                  CCC
3038  011130  000261                  SEC
3039  011132  005312                  DEC     (R2)            ;(R2)=077777,CC=0011
3040  011134  103001                  BCC     DEC1
3041  011136  003401                  BLE     .+4
3042  011140  104400            DEC1: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3043
3044  011142  000257                  CCC
3045  011144  000261                  SEC
3046  011146  005512                  ADC     (R2)            ;(R2)=100000,CC=1010
3047  011150  103403                  BCS     ADC1
3048  011152  102002                  BVC     ADC1
3049  011154  100001                  BPL     ADC1
3050  011156  001001                  BNE     .+4
3051  011160  104400            ADC1: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3052
3053  011162  006112                  ROL     (R2)            ;(R2)=000000,CC=0111
3054  011164  103003                  BCC     ROL1
3055  011166  102002                  BVC     ROL1
3056  011170  001001                  BNE     ROL1
3057  011172  100001                  BPL     .+4
3058  011174  104400            ROL1: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3059
3060  011176  006112                  ROL     (R2)            ;(R2)=000001,CC=0000
3061  011200  101402                  BLOS    ROL1A           ;BRANCH IF C OR Z IS SET
3062  011202  102401                  BVS     ROL1A
3063  011204  100001                  BPL     .+4
3064  011206  104400            ROL1A: HLT
3065
3066  011210  006212                  ASR     (R2)            ;(R2)=000000,CC=0111
3067  011212  103003                  BCC     ASR1
3068  011214  102002                  BVC     ASR1
3069  011216  001001                  BNE     ASR1
3070  011220  100001                  BPL     .+4
3071  011222  104400            ASR1: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3072
```

```
 3073  011224  006012                    ROR     (R2)            ;(R2)=100000,CC=1010
 3074  011226  103403                    BCS     ROR1A
 3075  011230  102002                    BVC     ROR1A
 3076  011232  001401                    BEQ     ROR1A
 3077  011234  100401                    BMI     .+4
 3078  011236  104400           ROR1A:   HLT
 3079
 3080  011240  000261                    SEC
 3081  011242  005212                    INC     (R2)            ;(R2)=100001,CC=1001
 3082  011244  103003                    BCC     INC1
 3083  011246  102402                    BVS     INC1
 3084  011250  001401                    BEQ     INC1
 3085  011252  100401                    BMI     .+4
 3086  011254  104400           INC1:    HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
 3087
 3088  011256  005612                    SBC     (R2)            ;(R2)=100000,CC=1000
 3089  011260  103403                    BCS     SBC1
 3090  011262  102402                    BVS     SBC1
 3091  011264  001401                    BEQ     SBC1
 3092  011266  100401                    BMI     .+4
 3093  011270  104400           SBC1:    HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
 3094
 3095  011272  000261                    SEC
 3096  011274  005612                    SBC     (R2)            ;(R2)=077777,CC=0010
 3097  011276  103403                    BCS     SBC1A
 3098  011300  102002                    BVC     SBC1A
 3099  011302  001401                    BEQ     SBC1A
 3100  011304  100001                    BPL     .+4
 3101  011306  104400           SBC1A:   HLT                     ;ERROR! INCORRECT CC'S AS SHOEN ABOVE
 3102
 3103  011310  000261                    SEC
 3104  011312  005512                    ADC     (R2)            ;(R2)=100000,CC=1010
 3105  011314  100401                    BMI     .+4
 3106  011316  104400                    HLT
 3107
 3108  011320  000261                    SEC
 3109  011322  006312                    ASL     (R2)            ;(R2)=000000,CC=0111
 3110  011324  103003                    BCC     ASL1
 3111  011326  102002                    BVC     ASL1
 3112  011330  001001                    BNE     ASL1
 3113  011332  100001                    BPL     .+4
 3114  011334  104400           ASL1:    HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
 3115
 3116  011336  005112                    COM     (R2)            ;(R2)=177777,CC=1001
 3117  011340  103002                    BCC     COM1
 3118  011342  102401                    BVS     COM1
 3119  011344  100401                    BMI     .+4
 3120  011346  104400           COM1:    HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
 3121
 3122  011350  000250                    CLN
 3123  011352  005712                    TST     (R2)            ;(R2)=177777,CC=1000
 3124  011354  103403                    BCS     TST1
 3125  011356  102402                    BVS     TST1
 3126  011360  100001                    BPL     TST1
 3127  011362  001001                    BNE     .+4
 3128  011364  104400           TST1:    HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
```

```
3129
3130  011366  000262                  SEV
3131  011370  005412                  NEG     (R2)            ;(R2)=000001,CC=0000
3132  011372  103002                  BCC     NEG1
3133  011374  102401                  BVS     NEG1
3134  011376  001001                  BNE     .+4
3135  011400  104400          NEG1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3136
3137  011402  005312                  DEC     (R2)            ;(R2)=000000,CC=0101
3138  011404  103001                  BCC     DEC1A
3139  011406  001401                  BEQ     .+4
3140  011410  104400          DEC1A:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3141  011412  104000                  SCOPE
3142
3143                           ;CHECK UNARY BYTE INSTRUCTIONS USING ADDRESS MODE 1
3144  011414  000401                  BR      .+4             ;RESERVE A WORD
3145  011416  000000                  .WORD   0               ;ADDRESS RESERVED FOR TESTS
3146  011420  010703                  MOV     PC,R3
3147  011422  162703  000004         SUB     #4,R3           ;R3 POINTS TO EVEN BYTE OF WORD
3148  011426  010304                  MOV     R3,R4           ;R4 POINTS TO ODD BYTE OF WORD
3149  011430  005204                  INC     R4
3150  011432  005013                  CLR     (R3)            ;PRESET DATA
3151
3152  011434  000261          1$:     SEC
3153  011436  105513                  ADCB    (R3)            ;ADD CARRY TO EVEN BYTE
3154  011440  100402                  BMI     2$              ;UNTIL EVEN BYTE BECOMES NEGATIVE
3155  011442  105214                  INCB    (R4)            ;INCREMENT ODD BYTE
3156  011444  000773                  BR      1$
3157  011446  102401          2$:     BVS     .+4             ;(R3)=077600=[0774][200],CC=1010
3158  011450  104400                  HLT
3159  011452  000242                  CLV
3160  011454  105214                  INCB    (R4)            ;(R3)=100200=[1000][200],CC=1010
3161  011456  103402                  BCS     INCB1
3162  011460  102001                  BVC     INCB1
3163  011462  100401                  BMI     .+4
3164  011464  104400          INCB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3165
3166  011466  106114                  ROLB    (R4)            ;(R3)=000200=[0000][200],CC=0111
3167  011470  103002                  BCC     ROLB1
3168  011472  102001                  BVC     ROLB1
3169  011474  001401                  BEQ     .+4
3170  011476  104400          ROLB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3171
3172  011500  105614                  SBCB    (R4)            ;(R3)=177600=[1774][200], CC=1001
3173  011502  103002                  BCC     SBCB1
3174  011504  102401                  BVS     SBCB1
3175  011506  100401                  BMI     .+4
3176  011510  104400          SBCB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3177
3178  011512  106313                  ASLB    (R3)            ;(R3)=177400,CC=0111
3179  011514  103002                  BCC     ASLB1
3180  011516  102001                  BVC     ASLB1
3181  011520  001401                  BEQ     .+4
3182  011522  104400          ASLB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3183
3184  011524  105413                  NEGB    (R3)            ;(R3)=177400,CC=0100
```

```
3185  011536  103402                  BCS     NEGB1
3186  011530  102401                  BVS     NEGB1
3187  011532  001401                  BEQ     .+4
3188  011534  104400          NEGB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3189
3190  011536  000277                  SCC
3191  011540  105313                  DECB    (R3)            ;(R3)=177777,CC=1001
3192  011542  103002                  BCC     DECB1
3193  011544  102401                  BVS     DECB1
3194  011546  001001                  BNE     .+4
3195  011550  104400          DECB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3196
3197  011552  000241                  CLC
3198  011554  106013                  RORB    (R3)            ;(R3)=177577,CC=0011
3199  011556  103002                  BCC     RORB1
3200  011560  102001                  BVC     RORB1
3201  011562  100001                  BPL     .+4
3202  011564  104400          RORB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3203
3204  011566  000241                  CLC
3205  011570  105114                  COMB    (R4)            ;(R3)=000177,CC=0101
3206  011572  103002                  BCC     COMB1
3207  011574  102401                  BVS     COMB1
3208  011576  001401                  BEQ     .+4
3209  011600  104400          COMB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3210
3211  011602  106213          1$:     ASRB    (R3)            ;SHIFT EVEN BYTE UNTIL V CLEARS
3212  011604  102002                  BVC     2$
3213  011606  105514                  ADCB    (R4)            ;AND ADD CARRY TO ODD BYTE
3214  011610  000774                  BR      1$
3215  011612  103401          2$:     BCS     ASRB1
3216  011614  001401                  BEQ     .+4
3217  011616  104400          ASRB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3218
3219  011620  106214                  ASRB    (R4)
3220  011622  106214                  ASRB    (R4)            ;(R3)=000400,CC=0011
3221  011624  103002                  BCC     ASRB1A
3222  011626  102001                  BVC     ASRB1A
3223  011630  001001                  BNE     .+4
3224  011632  104400          ASRB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3225
3226  011634  105314                  DECB    (R4)            ;(R3)=000000,CC=0100
3227  011636  001401                  BEQ     .+4
3228  011640  104400                  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3229
3230  011642  000261                  SEC
3231  011644  106014                  RORB    (R4)            ;(R3)=100000,CC=1010
3232  011646  103402                  BCS     RORB1A
3233  011650  102001                  BVC     RORB1A
3234  011652  100401                  BMI     .+4
3235  011654  104400          RORB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3236
3237  011656  000242                  CLV
3238  011660  105314                  DECB    (R4)            ;(R3)=077400,CC=0100
3239  011662  102401                  BVS     .+4
3240  011664  104400                  HLT
```

```
3241
3242  011666  000261                  SEC
3243  011670  105313                  DECB    (R3)            ;(R3)=077777,CC=1001
3244  011672  103002                  BCC     DECB1A
3245  011674  102401                  BVS     DECB1A
3246  011676  100401                  BMI     .+4
3247  011700  104400          DECB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3248
3249  011702  000277                  SCC
3250  011704  000313                  SWAB    (R3)            ;(R3)=177577=[1774][177],CC=0000
3251  011706  103402                  BCS     SWAB1
3252  011710  102401                  BVS     SWAB1
3253  011712  100001                  BPL     .+4
3254  011714  104400          SWAB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3255
3256  011716  105714                  TSTB    (R4)            ;(R3)=177577=[1774][177],CC=1000
3257  011720  103402                  BCS     TSTB1
3258  011722  102401                  BVS     TSTB1
3259  011724  100401                  BMI     .+4
3260  011726  104400          TSTB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3261
3262  011730  105014                  CLRB    (R4)            ;(R3)=000177=[0000][177],CC=0100
3263  011732  001401                  BEQ     .+4
3264  011734  104400                  HLT
3265  011736  106313                  ASLB    (R3)            ;(R3)=000376 ,CC=1010
3266  011740  103402                  BCS     ASLB1A
3267  011742  102001                  BVC     ASLB1A
3268  011744  100401                  BMI     .+4
3269  011746  104400          ASLB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3270
3271  011750  105113                  COMB    (R3)            ;(R3)=000001,CC=0001
3272  011752  103002                  BCC     COMB1A
3273  011754  102401                  BVS     COMB1A
3274  011756  100001                  BPL     .+4
3275  011760  104400          COMB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3276
3277  011762  000313                  SWAB    (R3)            ;(R3)=000400, CC=0100
3278  011764  001401                  BEQ     .+4
3279  011766  104400                  HLT
3280
3281  011770  105213                  INCB    (R3)
3282  011772  000261                  SEC
3283  011774  105613                  SBCB    (R3)            ;(R3)=000400,CC=0100
3284  011776  001401                  BEQ     .+4
3285  012000  104400                  HLT
3286  012002  022713  000400          CMP     #400,(R3)       ;CHECK REMAINING RESULT
3287  012006  001401                  BEQ     .+4
3288  012010  104400                  HLT
3289  012012  104000                  SCOPE
3290
3291                          ;CHECK UNARY WORD OPS USING ADDRESS MODES 2 AND 4 (AUTO INC/DEC)
3292  012014  000401                  BR      .+4
3293  012016  000000                  .WORD   0               ;ADDRESS RESERVED FOR TESTS
3294  012020  010704                  MOV     PC,R4
3295  012022  162704  000004          SUB     #4,R4           ;R4 AND R5 POINT TO
3296  012026  010405                  MOV     R4,R5           ;RESERVED WORD
```

```
3297  012030  005015                    CLR     (R5)            ;PRESET DATA=0
3298
3299  012032  000277                    SCC
3300  012034  000244                    CLZ
3301  012036  005725                    TST     (R5)+           ;(R5)=000000,CC=0100
3302  012040  103402                    BCS     TST2
3303  012042  102401                    BVS     TST2
3304  012044  001401                    BEQ     .+4
3305  012046  104400          TST2:     HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3306
3307  012050  005145                    COM     -(R5)           ;(R5)=177777,CC=1001
3308  012052  103001                    BCC     COM4
3309  012054  100401                    BMI     .+4
3310  012056  104400          COM4:     HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3311
3312  012060  000241                    CLC
3313  012062  006024                    ROR     (R4)+           ;(R4)=077777,CC=0011
3314  012064  103002                    BCC     ROR2
3315  012066  102001                    BVC     ROR2
3316  012070  100001                    BPL     .+4
3317  012072  104400          ROR2:     HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3318
3319  012074  000257                    CCC
3320  012076  005244                    INC     -(R4)           ;(R4)=100000,CC=1010
3321  012100  102002                    BVC     INC4
3322  012102  001401                    BEQ     INC4
3323  012104  100401                    BMI     .+4
3324  012106  104400          INC4:     HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3325
3326  012110  000261                    SEC
3327  012112  000324                    SWAB    (R4)+           ;(R4)=000200,CC=1000
3328  012114  103401                    BCS     SWAB2
3329  012116  100401                    BMI     .+4
3330  012120  104400          SWAB2:    HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3331
3332  012122  005425                    NEG     (R5)+           ;(R5)=177600,CC=1001
3333  012124  103001                    BCC     NEG2
3334  012126  100401                    BMI     .+4
3335  012130  104400          NEG2:     HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3336
3337  012132  005044                    CLR     -(R4)           ;(R4)=000000,CC=0100
3338  012134  001401                    BEQ     .+4
3339  012136  104400                    HLT
3340
3341  012140  000261                    SEC
3342  012142  006045                    ROR     -(R5)           ;(R5)=100000,CC=1010
3343  012144  000261                    SEC
3344  012146  005525                    ADC     (R5)+           ;(R5)=100001,CC=1000
3345  012150  102401                    BVS     ADC2
3346  012152  100401                    BMI     .+4
3347  012154  104400          ADC2:     HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3348
3349  012156  000262                    SEV
3350  012160  006224                    ASR     (R4)+           ;(R4)=140000,CC=1001
3351  012162  103002                    BCC     ASR2
3352  012164  102401                    BVS     ASR2
```

```
3353  012166  100401                  BMI     .+4
3354  012170  104400          ASR2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3355
3356  012172  000262                  SEV
3357  012174  006144                  ROL     -(R4)           ;(R4)=100001, CC=1001
3358  012176  103002                  BCC     ROL4
3359  012200  102401                  BVS     ROL4
3360  012202  100401                  BMI     .+4
3361  012204  104400          ROL4:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3362
3363  012206  005645                  SBC     -(R5)           ;(R5)=100000,CC=1000
3364  012210  103001                  BCC     .+4
3365  012212  104400                  HLT                     ;ERROR! 'C' BIT FAILED TO CLEAR
3366
3367  012214  005325                  DEC     (R5)+           ;(R5)=077777,CC=0010
3368  012216  103402                  BCS     DEC2
3369  012220  102001                  BVC     DEC2
3370  012222  100001                  BPL     .+4
3371  012224  104400          DEC2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3372
3373  012226  006324                  ASL     (R4)+           ;(R4)=177776,CC=1010
3374  012230  102401                  BVS     .+4
3375  012232  104400                  HLT
3376  012234  006344                  ASL     -(R4)           ;(R4)=177774,CC='001
3377  012236  103003                  BCC     ASL4
3378  012240  102402                  BVS     ASL4
3379  012242  001401                  BEQ     ASL4
3380  012244  100401                  BMI     .+4
3381  012246  104400          ASL4:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3382
3383  012250  022724  177774          CMP     #177774,(R4)+
3384  012254  001401                  BEQ     .+4
3385  012256  104400                  HLT
3386  012260  020405                  CMP     R4,R5
3387  012262  001401                  BEQ     .+4
3388  012264  104400                  HLT
3389  012266  104000                  SCOPE
3390
3391                          ;CHECK UNARY BYTE OPS USING ADDRESS MODES 2 AND 4
3392  012270  000401                  BR      .+4             ;RESERVE A WORD
3393  012272  000000                  .WORD   0               ;RESERVED WORD
3394  012274  010705                  MOV     PC,R5
3395  012276  162705  000004          SUB     #4,R5           ;R5 POINTS TO EVEN BYTE OF RESERVED WORD
3396  012302  010500                  MOV     R5,R0
3397  012304  010002                  MOV     R0,R2
3398  012306  005202                  INC     R2              ;R2 POINTS TO ODD BYTE OF RESERVED WORD
3399  012310  005010                  CLR     (R0)            ;PRESET
3400
3401  012312  000277                  SCC
3402  012314  000241                  CLC
3403  012316  105125                  COMB    (R5)+           ;(R0)=000377,CC=1001
3404  012320  103002                  BCC     COMB2
3405  012322  102401                  BVS     COMB2
3406  012324  100401                  BMI     .+4
3407  012326  104400          COMB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3408
```

```
3409  012330  105542                  ADCB    -(R2)           ;(R0)=000000,CC=0101
3410  012332  001401                  BEQ     .+4
3411  012334  104400                  HLT                     ;ERROR! INCORRECT RESULT AS SHOWN ABOVE
3412  012336  105525                  ADCB    (R5)+           ;(R0)=000400,CC=0000
3413  012340  103401                  BCS     ADCB2
3414  012342  001001                  BNE     .+4
3415  012344  104400          ADCB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3416
3417  012346  000263                  +SEC!SEV
3418  012350  106045                  RORB    -(R5)           ;(R0)=100000,CC=1001
3419  012352  103003                  BCC     RORB4
3420  012354  102402                  BVS     RORB4
3421  012356  001401                  BEQ     RORB4
3422  012360  100401                  BMI     .+4
3423  012362  104400          RORB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3424
3425  012364  000277                  SCC
3426  012366  106122                  ROLB    (R2)+           ;(R0)=100001,CC=0000
3427  012370  103403                  BCS     ROLB2
3428  012372  102402                  BVS     ROLB2
3429  012374  001401                  BEQ     ROLB2
3430  012376  100001                  BPL     .+4
3431  012400  104400          ROLB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3432
3433  012402  000257                  CCC
3434  012404  106225                  ASRB    (R5)+           ;(R0)=140001, CC=1010
3435  012406  103402                  BCS     ASRB2
3436  012410  102001                  BVC     ASRB2
3437  012412  100401                  BMI     .+4
3438  012414  104400          ASRB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3439
3440  012416  105242                  INCB    -(R2)           ;(R0)=140002,CC=0000
3441  012420  000277                  SCC
3442  012422  106222                  ASRB    (R2)+           ;(R0)=140001,CC=0000
3443  012424  103402                  BCS     ASRB2A
3444  012426  102401                  BVS     ASRB2A
3445  012430  100001                  BPL     .+4
3446  012432  104400          ASRB2A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3447
3448  012434  000266                  +SEZ!SEV                ;SET Z,V
3449  012436  106345                  ASLB    -(R5)           ;(R0)=100001,CC=1001
3450  012440  103003                  BCC     ASLB4
3451  012442  102402                  BVS     ASLB4
3452  012444  001401                  BEQ     ASLB4
3453  012446  100401                  BMI     .+4
3454  012450  104400          ASLB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3455
3456  012452  105322                  DECB    (R2)+           ;(R0)=077401=[0774][001] ,CC=0010
3457  012454  103002                  BCC     DECB2
3458  012456  102001                  BVC     DECB2
3459  012460  100001                  BPL     .+4
3460  012462  104400          DECB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWNABOVE
3461
3462  012464  105645                  SBCB    -(R5)           ;(R0)=077400, CC=0100
3463  012466  103402                  BCS     SBCB4
3464  012470  102401                  BVS     SBCB4
```

```
3465  012472  001401                  BEQ     .+4
3466  012474  104400          SBCB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3467
3468  012476  105442                  NEGB    -(R2)           ;(R0)=10400,CC=1001
3469  012500  103002                  BCC     NEGB4
3470  012502  102401                  BVS     NEGB4
3471  012504  100401                  BMI     .+4
3472  012506  104400          NEGB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
3473
3474  012510  105725                  TSTB    (R5)+           ;(R0)=100400,CC=0100
3475  012512  103401                  BCS     TSTB2
3476  012514  001401                  BEQ     .+4
3477  012516  104400          TSTB2:  HLT
3478
3479  012520  105722                  TSTB    (R2)+           ;(R0)=100400,CC=1000
3480  012522  001401                  BEQ     TSTB2A
3481  012524  100401                  BMI     .+4
3482  012526  104400          TSTB2A: HLT
3483
3484  012530  000261                  SEC
3485  012532  000342                  SWAB    -(R2)           ;(R0)=000201,CC=1000
3486  012534  103401                  BCS     SWAB4
3487  012536  100401                  BMI     .+4
3488  012540  104400          SWAB4:  HLT
3489
3490  012542  000277                  SCC
3491  012544  105225                  INCB    (R5)+           ;(R0)=000601=[0004][201],CC=0000
3492  012546  103003                  BCC     INCB2
3493  012550  102402                  BVS     INCB2
3494  012552  001401                  BEQ     INCB2
3495  012554  100001                  BPL     .+4
3496  012556  104400          INCB2:  HLT
3497
3498  012560  022227  000601          CMP     (R2)+,#000601   ;CHECK END RESULT
3499  012564  001401                  BEQ     .+4
3500  012566  104400                  HLT
3501  012570  020205                  CMP     R2,R5           ;CHECK REGISTERS
3502  012572  001401                  BEQ     .+4
3503  012574  104400                  HLT
3504  012576  104000                  SCOPE
3505
3506                          ;CHECK UNARY WORD OPS USING ADDRESS MODES 3 AND 5
3507  012600  000402                  BR      .+6             ;RESERVE 2 WORDS
3508  012602  000000                  .WORD   0               ;1 FOR THE ADDRESS
3509  012604  000000                  .WORD   0               ;AND 1 FOR DATA
3510  012606  010703                  MOV     PC,R3
3511  012610  162703  000004          SUB     #4,R3
3512  012614  005013                  CLR     (R3)            ;PRESET DATA
3513  012616  010300                  MOV     R3,R0           ;R0 POINTS TO DATA WORD
3514  012620  005743                  TST     -(R3)
3515  012622  010013                  MOV     R0,(R3)
3516  012624  010304                  MOV     R3,R4
3517
3518  012626  000257                  CCC
3519  012630  005733                  TST     @(R3)+          ;(R0)=000000,CC=0100
3520  012632  001401                  BEQ     .+4
```

```
3521  012634  104400                    HLT
3522
3523  012636  000261                    SEC
3524  012640  006053                    ROR     @-(R3)          ;(R0)=100000,CC=1010
3525  012642  103402                    BCS     RORS
3526  012644  102001                    BVC     RORS
3527  012646  100401                    BMI     .+4
3528  012650  104400              RORS: HLT
3529
3530  012652  000257                    CCC
3531  012654  006234                    ASR     @(R4)+          ;(R0)=140000,CC=1010
3532  012656  102001                    BVC     ASR3
3533  012660  100401                    BMI     .+4
3534  012662  104400              ASR3: HLT
3535
3536  012664  000250                    CLN
3537  012666  006333                    ASL     @(R3)+          ;(R0)=100000,CC=1001
3538  012670  103002                    BCC     ASL3
3539  012672  102401                    BVS     ASL3
3540  012674  100401                    BMI     .+4
3541  012676  104400              ASL3: HLT
3542
3543  012700  000277                    SCC
3544  012702  005354                    DEC     @-(R4)          ;(R0)=077777, CC=0010
3545  012704  103003                    BCC     DEC5
3546  012706  102002                    BVC     DEC5
3547  012710  001401                    BEQ     DEC5
3548  012712  100001                    BPL     .+4
3549  012714  104400              DEC5: HLT
3550
3551  012716  005453                    NEG     @-(R3)          ;(R0)=100001, CC=1001
3552  012720  103002                    BCC     NEG5
3553  012722  102401                    BVS     NEG5
3554  012724  100401                    BMI     .+4
3555  012726  104400              NEG5: HLT
3556
3557  012730  000262                    SEV
3558  012732  005134                    COM     @(R4)+          ;(R0)=077776, CC=0001
3559  012734  103001                    BCC     COM3
3560  012736  102001                    BVC     .+4
3561  012740  104400              COM3: HLT
3562
3563  012742  005233                    INC     @(R3)+          ;(R0)=077777, CC=0001
3564  012744  103001                    BCC     INC3
3565  012746  100001                    BPL     .+4
3566  012750  104400              INC3: HLT
3567
3568  012752  005554                    ADC     @-(R4)          ;(R0)=100000, CC=1010
3569  012754  103402                    BCS     ADC5
3570  012756  102001                    BVC     ADC5
3571  012760  100401                    BMI     .+4
3572  012762  104400              ADC5: HLT
3573
3574  012764  000257                    CCC
3575  012766  006134                    ROL     @(R4)+          ;(R0)=000000,CC=0111
3576  012770  103002                    BCC     ROL3
```

```
3577  012772  102001                    BVC     ROL3
3578  012774  001401                    BEQ     .+4
3579  012776  104400            ROL3:   HLT
3580
3581  013000  005253                    INC     @-(R3)          ;(R0)=000001, CC=0001
3582  013002  005654                    SBC     @-(R4)          ;(R0)=000000, CC=0100
3583  013004  103401                    BCS     SBC5
3584  013006  001401                    BEQ     .+4
3585  013010  104400            SBC5:   HLT
3586  013012  104000                    SCOPE
3587
3588                            ;CHECK UNARY BYTE OPS USING ADDRESS MODES 3 AND 5
3589  013014  000403                    BR      .+10            ;RESERVE 3 WORDS
3590  013016  000000                    .WORD   0               ;1 FOR EVEN BYTE ADDRESS
3591  013020  000000                    .WORD   0               ;1 FOR ODD BYTE ADDRESS
3592  013022  000000                    .WORD   0               ;AND 1 FOR DATA
3593  013024  010702                    MOV     PC,R2
3594  013026  005742                    TST     -(R2)           ;BACK R2 UP TO
3595  013030  005742                    TST     -(R2)           ;DATA WORD
3596  013032  010200                    MOV     R2,R0           ;R0 POINTS TO THE DATA WORD
3597  013034  005010                    CLR     (R0)            ;PRESET DATA
3598  013036  005742                    TST     -(R2)           ;BACK R2 UP TO
3599  013040  005742                    TST     -(R2)           ;EVEN BYTE ADDRESS WORD
3600  013042  010022                    MOV     R0,(R2)+        ;LOAD ADDRESS
3601  013044  005200                    INC     R0              ;ODD BYTE ADDRESS
3602  013046  010022                    MOV     R0,(R2)+        ;LOAD ODD BYTE ADDRESS
3603  013050  010200                    MOV     R2,R0           ;RESET R0
3604  013052  010205                    MOV     R2,R5
3605
3606  013054  105152                    COMB    @-(R2)          ;(R0)=177400,CC=1001
3607  013056  103001                    BCC     COMB5
3608  013060  100401                    BMI     .+4
3609  013062  104400            COMB5:  HLT
3610
3611  013064  105752                    TSTB    @-(R2)          ;(R0)=177400, CC=0100
3612  013066  001401                    BEQ     .+4
3613  013070  104400                    HLT
3614
3615  013072  000262                    SEV
3616  013074  106255                    ASRB    @-(R5)          ;(R0)=177400, CC=1001
3617  013076  103002                    BCC     ASRB5
3618  013100  102401                    BVS     ASRB5
3619  013102  100401                    BMI     .+4
3620  013104  104400            ASRB5:  HLT
3621
3622  013106  105232                    INCB    @(R2)+          ;(R0)=177401, CC=000
3623  013110  103001                    BCC     INCB3
3624  013112  100001                    BPL     .+4
3625  013114  104400            INCB3:  HLT
3626
3627  013116  000241                    CLC
3628  013120  106055                    RORB    @-(R5)          ;(R0)=177400, CC=0111
3629  013122  103003                    BCC     RORB5
3630  013124  102002                    BVC     RORB5
3631  013126  001001                    BNE     RORB5
3632  013130  100001                    BPL     .+4
```

```
  3633  013132  104400                RORB5:  HLT
  3634
  3635  013134  106332                        ASLB    @(R2)+          ;(R0)=177000, CC=1001
  3636  013136  103002                        BCC     ASLB3
  3637  013140  102401                        BVS     ASLB3
  3638  013142  100401                        BMI     .+4
  3639  013144  104400                ASLB3:  HLT
  3640
  3641  013146  105552                        ADCB    @-(R2)          ;(R0)=177400, CC=1000
  3642  013150  103401                        BCS     ADCB5
  3643  013152  100401                        BMI     .+4
  3644  013154  104400                ADCB5:  HLT
  3645
  3646  013156  000277                        SCC
  3647  013160  106135                        ROLB    @(R5)+          ;(R0)=177401, CC=0000
  3648  013162  101402                        BLOS    ROLB3           ;BRANCH IF C OR Z IS SET
  3649  013164  102401                        BVS     ROLB3
  3650  013166  100001                        BPL     .+4
  3651  013170  104400                ROLB3:  HLT
  3652
  3653  013172  000352                        SWAB    @-(R2)          ;(R0)=000777, CC=1000
  3654  013174  100401                        BMI     .+4
  3655  013176  104400                        HLT
  3656
  3657  013200  000261                        SEC
  3658  013202  105635                        SBCB    @(R5)+          ;(R0)=000377, CC=0100
  3659  013204  103401                        BCS     SBCB3
  3660  013206  001401                        BEQ     .+4
  3661  013210  104400                SBCB3:  HLT
  3662
  3663  013212  105432                        NEGB    @(R2)+          ;(R0)=000001
  3664  013214  105352                        DECB    @-(R2)          ;(R0)=000000, CC=0101
  3665  013216  103001                        BCC     DECB5
  3666  013220  001401                        BEQ     .+4
  3667  013222  104400                DECB5:  HLT
  3668  013224  104000                        SCOPE
  3669
  3670                                ;CHECK UNARY WORD OPS USING ADDRESS MODE 6 (PC)
  3671  013226  005027                        CLR     (PC)+           ;PRESET DATA = 0
  3672  013230  000000                UWM6:   .WORD   0               ;RESERVED FOR DATA
  3673  013232  010700                        MOV     PC,R0
  3674  013234  024040                        CMP     -(R0),-(R0)     ;R0 POINTS TO DATA WORD
```

```
 3675  013236  000277                        SCC
 3676  013240  006167  177764                ROL     UWM6            ;(R0)=000001,CC=0000
 3677  013244  103403                        BCS     ROL6
 3678  013246  102402                        BVS     ROL6
 3679  013250  001401                        BEQ     ROL6
 3680  013252  100001                        BPL     .+4
 3681  013254  104400              ROL6:     HLT
 3682
 3683  013256  005167  177746                COM     UWM6            ;(R0)=177776, CC=1001
 3684  013262  103002                        BCC     COM6
 3685  013264  102401                        BVS     COM6
 3686  013266  100401                        BMI     .+4
 3687  013270  104400              COM6:     HLT
 3688  013272  006267  177732                ASR     UWM6            ;(R0)=177777, CC=1010
 3689  013276  103402                        BCS     ASR6
 3690  013300  102001                        BVC     ASR6
 3691  013302  100401                        BMI     .+4
 3692  013304  104400              ASR6:     HLT
 3693
 3694  013306  000277                        SCC
 3695  013310  005467  177714                NEG     UWM6            ;(R0)=000001, CC=0001
 3696  013314  103003                        BCC     NEG6
 3697  013316  102402                        BVS     NEG6
 3698  013320  001401                        BEQ     NEG6
 3699  013322  100001                        BPL     .+4
 3700  013324  104400              NEG6:     HLT
 3701
 3702  013326  000277                        SCC
 3703  013330  006067  177674                ROR     UWM6            ;(R0)=100000, CC=1001
 3704  013334  103003                        BCC     ROR6
 3705  013336  102402                        BVS     ROR6
 3706  013340  001401                        BEQ     ROR6
 3707  013342  100401                        BMI     .+4
 3708  013344  104400              ROR6:     HLT
 3709
 3710  013346  005667  177656                SBC     UWM6            ;(R0)=077777, CC=0010
 3711  013352  103402                        BCS     SBC6
 3712  013354  102001                        BVC     SBC6
 3713  013356  100001                        BPL     .+4
 3714  013360  104400              SBC6:     HLT
 3715
 3716  013362  000242                        CLV
 3717  013364  005267  177640                INC     UWM6            ;(R0)=100000, CC=1011
 3718  013370  103403                        BCS     INC6
 3719  013372  102002                        BVC     INC6
 3720  013374  001401                        BEQ     INC6
 3721  013376  100401                        BMI     .+4
 3722  013400  104400              INC6:     HLT
 3723
 3724  013402  006267  177622                ASR     UWM6            ;(R0)=140000, CC=1010
 3725  013406  000261                        SEC
 3726  013410  006367  177614                ASL     UWM6            ;(R0)=100000, CC=1001
 3727  013414  103002                        BCC     ASL6
 3728  013416  102401                        BVS     ASL6
 3729  013420  100401                        BMI     .+4
 3730  013422  104400              ASL6:     HLT
```

```
3731
3732  013424  005367  177600          DEC     UWM6            ;(R0)=077777, CC=0011
3733  013430  103002                  BCC     DEC6
3734  013432  102001                  BVC     DEC6
3735  013434  100001                  BPL     .+4
3736  013436  104400          DEC6:   HLT
3737
3738  013440  005567  177564          ADC     UWM6            ;(R0)=100000, CC=1010
3739  013444  103402                  BCS     ADC6
3740  013446  102001                  BVC     ADC6
3741  013450  100401                  BMI     .+4
3742  013452  104400          ADC6:   HLT
3743  013454  000242                  CLV
3744  013456  000367  177546          SWAB    UWM6
3745  013462  100401                  BMI     .+4
3746  013464  104400                  HLT
3747  013466  022710  000200          CMP     #200,(R0)
3748  013472  001401                  BEQ     .+4
3749  013474  104400                  HLT
3750  013476  104000                  SCOPE
3751
3752                          ;CHECK UNARY BYTE OPS (EVEN/ODD) USING ADDRESS MODE 6 (PC)
3753  013500  012700  014042          MOV     #UBM6,R0
3754  013504  063700  001004          ADD     @#FACTOR,R0     ;R0 POINTS TO ADDRESS OF DATA
3755  013510  005067  000326          CLR     UBM6            ;CLEAR DATA
3756  013514  000277                  SCC
3757  013516  000244                  CLZ
3758  013520  105767  000316          TSTB    UBM6
3759  013524  103403                  BCS     TSTB6
3760  013526  102402                  BVS     TSTB6
3761  013530  001001                  BNE     TSTB6
3762  013532  100001                  BPL     .+4
3763  013534  104400          TSTB6:  HLT
3764
3765  013536  000257                  CCC
3766  013540  105767  000277          TSTB    UBM6+1          ;TEST ODD BYTE
3767  013544  001401                  BEQ     .+4
3768  013546  104400                  HLT
3769
3770  013550  105667  000266          SBCB    UBM6            ;(R0)=000000, CC=0100
3771  013554  103402                  BCS     SBCB6
3772  013556  102401                  BVS     SBCB6
3773  013560  001401                  BEQ     .+4
3774  013562  104400          SBCB6:  HLT
3775
3776  013564  000261          1$:     SEC
3777  013566  105267  000250          INCB    UBM6            ;LOOP UNTIL (R0)=077600, CC=1011
3778  013572  100403                  BMI     2$
3779  013574  105567  000243          ADCB    UBM6+1          ;INCB INST INCREMENTS EVEN BYTE
3780  013600  000771                  BR      1$              ;ADCB INCREMENTS ODD BYTE
3781  013602  103001          2$:     BCC     INCB6
3782  013604  102401                  BVS     .+4
3783  013606  104400          INCB6:  HLT
3784
3785  013610  106367  000226          ASLB    UBM6            ;(R0)=077400, CC=0111
3786  013614  103003                  BCC     ASLB6
```

```
3787  013616  102002                        BVC     ASLB6
3788  013620  001001                        BNE     ASLB6
3789  013622  100001                        BPL     .+4
3790  013624  104400              ASLB6:    HLT
3791
3792  013626  000242                        CLV
3793  013630  105567  000207                ADCB    UBM6+1          ;(R0)=100000, CC=1010
3794  013634  103402                        BCS     ADCB6
3795  013636  102001                        BVC     ADCB6
3796  013640  100401                        BMI     .+4
3797  013642  104400              ADCB6:    HLT
3798
3799  013644  000261                        SEC
3800  013646  106067  000171                RORB    UBM6+1          ;(R0)=140000, CC=1010
3801  013652  103402                        BCS     RORB6
3802  013654  102001                        BVC     RORB6
3803  013656  100401                        BMI     .+4
3804  013660  104400              RORB6:    HLT
3805
3806  013662  105167  000154                COMB    UBM6            ;(R0)=140377 CC=1001
3807  013666  103002                        BCC     COMB6
3808  013670  102401                        BVS     COMB6
3809  013672  100401                        BMI     .+4
3810  013674  104400              COMB6:    HLT
3811
3812  013676  000262                        SEV
3813  013700  105467  000137                NEGB    UBM6+1          ;(R0)=040377, CC=0001
3814  013704  103002                        BCC     NEGB6
3815  013706  102401                        BVS     NEGB6
3816  013710  100001                        BPL     .+4
3817  013712  104400              NEGB6:    HLT
3818
3819  013714  106167  000123                ROLB    UBM6+1          ;(R0)=100777, CC=1010
3820  013720  103402                        BCS     ROLB6
3821  013722  102001                        BVC     ROLB6
3822  013724  100401                        BMI     .+4
3823  013726  104400              ROLB6:    HLT
3824
3825  013730  106267  000106                ASRB    UBM6            ;(R0)=100777, CC=1001
3826  013734  103002                        BCC     ASRB6
3827  013736  102401                        BVS     ASRB6
3828  013740  100401                        BMI     .+4
3829  013742  104400              ASRB6:    HLT
3830
3831  013744  105267  000072                INCB    UBM6            ;(R0)=100400, CC=0101
3832  013750  103002                        BCC     INCB6A
3833  013752  102401                        BVS     INCB6A
3834  013754  001401                        BEQ     .+4
3835  013756  104400              INCB6A:   HLT
3836
3837  013760  105367  000057                DECB    UBM6+1          ;(R0)=100000, CC=1001
3838  013764  103003                        BCC     DECB6A
3839  013766  102402                        BVS     DECB6A
3840  013770  001401                        BEQ     DECB6A
3841  013772  100401                        BMI     .+4
3842  013774  104400              DECB6A:   HLT
```

```
3843
3844  013776  000367  000040                SWAB    UBM6            ;(R0)=000200, CC=1000
3845  014002  103401                        BCS     SWAB6
3846  014004  100401                        BMI     .+4
3847  014006  104400              SWAB6:    HLT
3848
3849  014010  106167  000026                ROLB    UBM6            ;(R0)=000000, CC=0111
3850  014014  103002                        BCC     ROLB6A
3851  014016  102001                        BVC     ROLB6A
3852  014020  001401                        BEQ     .+4
3853  014022  104400              ROLB6A:   HLT
3854
3855  014024  005767  000012                TST     UBM6            ;(R0)=000000, CC=0100
3856  014030  103402                        BCS     TST6
3857  014032  102401                        BVS     TST6
3858  014034  001401                        BEQ     .+4
3859  014036  104400              TST6:     HLT
3860
3861  014040  000401                        BR      .+4             ;RESERVE A WORD
3862  014042  000000              UBM6:     .WORD   0               ;WORD RESERVED FOR DATA
3863  014044  104000                        SCOPE
3864  014046  010702                        MOV     PC,R2
3865  014050  062702  000012                ADD     #12,R2
3866  014054  012707  001152                MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
3867  014060  000000              REL11:    .WORD   0
3868                              ;11111111111111 LAST ADDRESS OF CODE TO BE RELOCATED 11111111111
3869
3870
3871                                        .SBTTL  START OF SECTION 2
3872                              ;2222222222222 FIRST ADDRESS TO BE RELOCATED 222222222
3873  014062  010700              REL2:     MOV     PC,R0           ;GET PC
3874  014064  005740                        TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL2
3875  014066  010037  001010                MOV     R0,@#FRSTAD     ;SAVE
3876  014072  012737  000002  005442        MOV     #2,@#SECT       ;SET SECTION #
3877  014100  004737  005432                JSR     PC,@#LDDISP     ;LOAD DISPLAY GEG
3878  014104  013767  005436  003744        MOV     @#DISPLY,REL22
3879  014112  010700                        MOV     PC,R0           ;GET CURRENT PC
3880  014114  162700  014114                SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
3881  014120  010037  001004                MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
3882  014124  010701                        MOV     PC,R1           ;SET NEW SCOPE PTR
3883
3884                              ;CHECK UNARY WORD OPS USING ADDRESS MODE 7
3885  014126  000403                        BR      UW7             ;RESERVE 3 WORDS FOR ADDRESSES & DATA
3886  014130  000000                        .WORD   0               ;CONTAINS ADDRESS OF UWM7
3887  014132  000000              UWM7:     .WORD   0               ;CONTAINS DATA
3888  014134  000000                        .WORD   0               ;CONTAINS ADDRESS OF UWM7
3889
3890  014136  010700              UW7:      MOV     PC,R0
3891  014140  005740                        TST     -(R0)
3892  014142  005740                        TST     -(R0)
3893  014144  005040                        CLR     -(R0)           ;CLEAR TEST DATA
3894  014146  010002                        MOV     R0,R2
3895  014150  010240                        MOV     R2,-(R0)        ;SET UP ADDRESS
3896  014152  005720                        TST     (R0)+           ;MOVE R0 TO NEXT ADDRESS
3897  014154  005720                        TST     (R0)+
3898  014156  010210                        MOV     R2,(R0)         ;SET NEXT ADDRESS
```

```
3899  014160  010200                    MOV     R2,R0           ;SET R0 POINTING TO DATA
3900  014162  000277                    SCC
3901  014164  000244                    CLZ
3902  014166  005772  000002            TST     @2(2)           ;(R0)=000000, CC=0100
3903  014172  001401                    BEQ     .+4
3904  014174  104400                    HLT
3905
3906  014176  000277                    SCC
3907  014200  005672  177776            SBC     @-2(2)          ;(R0)=177777, CC=1001
3908  014204  103002                    BCC     SBC7
3909  014206  102401                    BVS     SBC7
3910  014210  100401                    BMI     .+4
3911  014212  104400            SBC7:   HLT
3912
3913  014214  000277                    SCC
3914  014216  000241                    CLC
3915  014220  006372  000002            ASL     @2(2)           ;(R0)=177776, CC=1001
3916  014224  103002                    BCC     ASL7
3917  014226  102401                    BVS     ASL7
3918  014230  100401                    BMI     .+4
3919  014232  104400            ASL7:   HLT
3920
3921  014234  000257                    CCC
3922  014236  005372  000002            DEC     @2(2)           ;(R0)=177775, CC=1000
3923  014242  103402                    BCS     DEC7
3924  014244  102401                    BVS     DEC7
3925  014246  100401                    BMI     .+4
3926  014250  104400            DEC7:   HLT
3927
3928  014252  000262                    SEV
3929  014254  006272  177776            ASR     @-2(2)          ;(R0)=177776, CC=1001
3930  014260  103002                    BCC     ASR7
3931  014262  102401                    BVS     ASR7
3932  014264  100401                    BMI     .+4
3933  014266  104400            ASR7:   HLT
3934
3935  014270  000241                    CLC
3936  014272  000262                    SEV
3937  014274  006072  177776            ROR     @-2(2)          ;(R0)=077777, CC=0000
3938  014300  101402                    BLOS    ROR7            ;BRANCH IF C OR Z IS SET
3939  014302  102401                    BVS     ROR7
3940  014304  100001                    BPL     .+4
3941  014306  104400            ROR7:   HLT
3942
3943  014310  000262                    SEV
3944  014312  005472  000002            NEG     @2(2)           ;(R0)=100001, CC=1001
3945  014316  103002                    BCC     NEG7
3946  014320  102401                    BVS     NEG7
3947  014322  100401                    BMI     .+4
3948  014324  104400            NEG7:   HLT
3949
3950  014326  000250                    CLN
3951  014330  000372  177776            SWAB    @-2(2)          ;(R0)=000600, CC=1000
3952  014334  103401                    BCS     SWAB7
3953  014336  100401                    BMI     .+4
3954  014340  104400            SWAB7:  HLT
```

```
3955
3956  014342  000262                          SEV
3957  014344  005172  000002                  COM     @2(2)           ;(R0)=177177, CC=1001
3958  014350  103002                          BCC     COM7
3959  014352  102401                          BVS     COM7
3960  014354  100401                          BMI     .+4
3961  014356  104400                  COM7:   HLT
3962
3963  014360  000372  000002                  SWAB    @2(2)           ;(R0)=077776, CC=1000
3964  014364  100401                          BMI     .+4
3965  014366  104400                          HLT
3966
3967  014370  000277                          SCC
3968  014372  005572  177776                  ADC     @-2(2)          ;(R0)=077777, CC=0000
3969  014376  103402                          BCS     ADC7
3970  014400  102401                          BVS     ADC7
3971  014402  100001                          BPL     .+4
3972  014404  104400                  ADC7:   HLT
3973
3974  014406  005272  000002                  INC     @2(2)           ;(R0)=100000, CC=1010
3975  014412  102001                          BVC     INC7
3976  014414  100401                          BMI     .+4
3977  014416  104400                  INC7:   HLT
3978
3979  014420  000257                          CCC
3980  014422  006172  177776                  ROL     @-2(2)          ;(R0)=000000, CC=0111
3981  014426  103002                          BCC     ROL7
3982  014430  102001                          BVC     ROL7
3983  014432  001401                          BEQ     .+4
3984  014434  104400                  ROL7:   HLT
3985  014436  104000                          SCOPE
3986
3987                                  ;CHECK UNARY BYTE OPS USING ADDRESS MODE 7
3988  014440  005720                          TST     (R0)+
3989  014442  005210                          INC     (R0)            ;WORD FOLLOWING UNM7 CONTAINS ADDRESS
3990  014444  005740                          TST     -(R0)           ;OF ODD BYTE, R0 POINTS TO DATA WORD
3991  014446  005010                          CLR     (R0)            ;PRESET DATA
3992  014450  010701                          MOV     PC,R1           ;SET SCOPE PTR
3993                                  ;NOTE:  @2(2) REFERENCES THE ODD BYTE, AND @-2(2) REFERENCES THE EVEN BYTE.
3994
3995  014452  000263                          +SEC!SEV                ;SET C AND V
3996  014454  105672  000002                  SBCB    @2(2)           ;(R0)=177400, CC=1001
3997  014460  103003                          BCC     SBCB7
3998  014462  102402                          BVS     SBCB7
3999  014464  001401                          BEQ     SBCB7
4000  014466  100401                          BMI     .+4
4001  014470  104400                  SBCB7:  HLT
4002
4003  014472  000277                          SCC                     ;SET CONDITION CODES
4004  014474  105572  :77776                  ADCB    @-2(2)          ;(R0)=177401, CC=0000
4005  014500  103403                          BCS     ADCB7
4006  014502  102402                          BVS     ADCB7
4007  014504  001401                          BEQ     ADCB7
4008  014506  100001                          BPL     .+4
4009  014510  104400                  ADCB7:  HLT
4010
```

```
4011  014512  105172  177776          COMB    @-2(2)          ;(R0)=177776, CC=1001
4012  014516  103002                  BCC     COMB7
4013  014520  102401                  BVS     COMB7
4014  014522  100401                  BMI     .+4
4015  014524  104400          COMB7:  HLT
4016
4017  014526  000241                  CLC                     ;CLEAR CARRY
4018  014530  106072  000002          RORB    @2(2)           ;(R0)=077776, CC=0011
4019  014534  103002                  BCC     RORB7
4020  014536  102001                  BVC     RORB7
4021  014540  100001                  BPL     .+4
4022  014542  104400          RORB7:  HLT
4023
4024  014544  105272  000002          INCB    @2(2)           ;(R0)=100376, CC=1011
4025  014550  103002                  BCC     INCB7
4026  014552  102001                  BVC     INCB7
4027  014554  100401                  BMI     .+4
4028  014556  104400          INCB7:  HLT
4029
4030  014560  105372  177776          DECB    @-2(2)          ;(R0)=100375, CC=1001
4031  014564  103002                  BCC     DECB7
4032  014566  102401                  BVS     DECB7
4033  014570  100401                  BMI     .+4
4034  014572  104400          DECB7:  HLT
4035
4036  014574  106372  000002          ASLB    @2(2)           ;(R0)=000375, CC=0111
4037  014600  103002                  BCC     ASLB7
4038  014602  102001                  BVC     ASLB7
4039  014604  001401                  BEQ     .+4
4040  014606  104400          ASLB7:  HLT
4041
4042  014610  000241                  CLC                     ;CLEAR CARRY
4043  014612  106272  177776          ASRB    @-2(2)          ;(R0)=000376, CC=1001
4044  014616  103002                  BCC     ASRB7
4045  014620  102401                  BVS     ASRB7
4046  014622  100401                  BMI     .+4
4047  014624  104400          ASRB7:  HLT
4048
4049  014626  105472  000002          NEGB    @2(2)           ;(R0)=000376, CC=0100
4050  014632  103402                  BCS     NEGB7
4051  014634  102401                  BVS     NEGB7
4052  014636  001401                  BEQ     .+4
4053  014640  104400          NEGB7:  HLT
4054
4055  014642  000262                  SEV
4056  014644  106172  177776          ROLB    @-2(2)          ;(R0)=00374, CC=1001
4057  014650  103002                  BCC     ROLB7
4058  014652  102401                  BVS     ROLB7
4059  014654  100401                  BMI     .+4
4060  014656  104400          ROLB7:  HLT
4061
4062  014660  105272  177776          INCB    @-2(2)          ;(R0)=000375, CC=1001
4063  014664  105272  177776          INCB    @-2(2)          ;(R0)=000376, CC=1001
4064  014670  105572  177776          ADCB    @-2(2)          ;(R0)=000377, CC=1000
4065  014674  105172  177776          COMB    @-2(2)          ;(R0)=000000, CC=0100
4066  014700  001401                  BEQ     .+4
```

```
4067  014702  104400                    HLT
4068  014704  104000                    SCOPE
4069
4070                                    ;CHECK BINARY OPS  USING ADDRESS MODE 0
4071  014706  000277                    SCC                         ;SET CONDITION CODES
4072  014710  010700                    MOV     PC,R0               ;R0=PC, CC=X001
4073  014712  103002                    BCC     MOV0
4074  014714  102401                    BVS     MOV0
4075  014716  001001                    BNE     .+4
4076  014720  104400            MOV0:   HLT
4077
4078  014722  010002                    MOV     R0,R2               ;R2=R0
4079  014724  000262                    SEV                         ;SET V
4080  014726  160002                    SUB     R0,R2               ;R2=000000, CC=0100
4081  014730  103402                    BCS     SUB0
4082  014732  102401                    BVS     SUB0
4083  014734  001401                    BEQ     .+4
4084  014736  104400            SUB0:   HLT
4085
4086  014740  000244                    CLZ
4087  014742  010203                    MOV     R2,R3               ;R2=R3=000000, CC=0100
4088  014744  103401                    BCS     MOV0A
4089  014746  001401                    BEQ     .+4
4090  014750  104400            MOV0A:  HLT
4091
4092  014752  000257                    CCC
4093  014754  000272                    +SEV!SEN                    ;SET V & N
4094  014756  020203                    CMP     R2,R3               ;R2=R3=000000, CC=0100
4095  014760  103403                    BCS     CMP0
4096  014762  102402                    BVS     CMP0
4097  014764  001001                    BNE     CMP0
4098  014766  100001                    BPL     .+4
4099  014770  104400            CMP0:   HLT
4100
4101  014772  010002                    MOV     R0,R2               ;R0=R2
4102  014774  010203                    MOV     R2,R3               ;R0=R2=R3
4103  014776  060203                    ADD     R2,R3               ;R3=2*R0
4104  015000  006302                    ASL     R2                  ;R2=2*R0
4105  015002  020203                    CMP     R2,R3               ;R2=R3=2*R0
4106  015004  001401                    BEQ     .+4
4107  015006  104400                    HLT                         ;ERROR! CHECK ADD INSTRUCTION
4108
4109                                    ;THE FOLLOWING SUBTEST SHIFTS A BIT THROUGH R2 AND R5 AND DOES A
4110                                    ;BIT TEST (BIT) USING R2 AND R5.
4111  015010  005002                    CLR     R2
4112  015012  005202                    INC     R2
4113  015014  000402                    BR      2$
4114  015016  006302            1$:     ASL     R2
4115  015020  100407                    BMI     4$
4116  015022  010205            2$:     MOV     R2,R5
4117  015024  000277                    SCC
4118  015026  030205                    BIT     R2,R5               ;R2=R5
4119  015030  103002                    BCC     3$
4120  015032  102401                    BVS     3$
4121  015034  001370                    BNE     1$
4122  015036  104400            3$:     HLT
```

```
4123  015040  010205          4$:     MOV     R2,R5
4124  015042  000257                  CCC
4125  015044  030205                  BIT     R2,R5
4126  015046  100401                  BMI     .+4
4127  015050  104400                  HLT
4128
4129  015052  005002                  CLR     R2
4130  015054  000277                  SCC
4131  015056  050002                  BIS     R0,R2
4132  015060  103002                  BCC     BISO
4133  015062  102401                  BVS     BISO
4134  015064  001001                  BNE     .+4
4135  015066  104400          BISO:   HLT
4136
4137  015070  010003                  MOV     R0,R3
4138  015072  000277                  SCC
4139  015074  000244                  CLZ
4140  015076  040003                  BIC     R0,R3
4141  015100  103003                  BCC     BICO
4142  015102  102402                  BVS     BICO
4143  015104  001001                  BNE     BICO
4144  015106  100001                  BPL     .+4
4145  015110  104400          BICO:   HLT
4146
4147  015112  010004                  MOV     R0,R4
4148  015114  005104                  COM     R4
4149  015116  040004                  BIC     R0,R4
4150  015120  005104                  COM     R4
4151  015122  020004                  CMP     R0,R4
4152  015124  001401                  BEQ     .+4
4153  015126  104400                  HLT
4154
4155  015130  010004                  MOV     R0,R4
4156  015132  005104                  COM     R4
4157  015134  010403                  MOV     R4,R3
4158  015136  050003                  BIS     R0,R3
4159  015140  103001                  BCC     BISOA
4160  015142  100401                  BMI     .+4
4161  015144  104400          BISOA:  HLT
4162  015146  005203                  INC     R3
4163  015150  001401                  BEQ     .+4
4164  015152  104400                  HLT
4165  015154  010304                  MOV     R3,R4           ;R3=R4=0
4166  015156  005103                  COM     R3              ;R3=177777
4167  015160  000261                  SEC                     ;SET C
4168  015162  006004                  ROR     R4              ;R4=100000
4169  015164  060304                  ADD     R3,R4           ;R3=177777,R4=077777, CC=0011
4170  015166  103003                  BCC     ADDO
4171  015170  102002                  BVC     ADDO
4172  015172  001401                  BEQ     ADDO
4173  015174  100001                  BPL     .+4
4174  015176  104400          ADDO:   HLT
4175  015200  010700                  MOV     PC,R0
4176  015202  022020                  CMP     (R0)+,(R0)+
4177  015204  020007                  CMP     R0,PC
4178  015206  001401                  BEQ     .+4
```

```
4179  015210  104400                  HLT
4180
4181  015212  010700                  MOV     PC,R0
4182  015214  062700  000010          ADD     #10,R0
4183  015220  010002                  MOV     R0,R2
4184  015222  020700                  CMP     PC,R0
4185  015224  001002                  BNE     CMPOA
4186  015226  020200                  CMP     R2,R0
4187  015230  001401                  BEQ     .+4
4188  015232  104400          CMPOA:  HLT
4189  015234  104000                  SCOPE
4190
4191
4192                          ;CHECK BINARY OPS USING ADDRESS MODE 1
4193  015236  000402                  BR      .+6             ;RESERVE TWO WORDS
4194  015240  000000                  .WORD   0               ;RESERVED FOR SOURCE DATA
4195  015242  000000                  .WORD   0               ;RESERVED FOR DESTINATION DATA
4196  015244  010704                  MOV     PC,R4
4197  015246  005744                  TST     -(R4)
4198  015250  005044                  CLR     -(R4)           ;R4 POINTS TO DESTINATION DATA
4199  015252  010403                  MOV     R4,R3
4200  015254  005043                  CLR     -(R3)           ;R3 POINTS TO SOUCE DATA
4201
4202  015256  005113                  COM     (R3)            ;(R3)=177777
4203  015260  005214                  INC     (R4)            ;(R4)=000001
4204  015262  000262                  SEV                     ;SET V
4205  015264  061314                  ADD     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=0101
4206  015266  103002                  BCC     ADD1
4207  015270  102401                  BVS     ADD1
4208  015272  001401                  BEQ     .+4
4209  015274  104400          ADD1:   HLT
4210
4211  015276  000277                  SCC
4212  015300  000250                  CLN
4213  015302  021314                  CMP     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=1000
4214  015304  103403                  BCS     CMP1
4215  015306  102402                  BVS     CMP1
4216  015310  001401                  BEQ     CMP1
4217  015312  100401                  BMI     .+4
4218  015314  104400          CMP1:   HLT
4219
4220  015316  000277                  SCC
4221  015320  000244                  CLZ
4222  015322  031314                  BIT     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=0101
4223  015324  103002                  BCC     BIT1
4224  015326  102401                  BVS     BIT1
4225  015330  001401                  BEQ     .+4
4226  015332  104400          BIT1:   HLT
4227
4228  015334  000277                  SCC
4229  015336  000245                  +CLC!CLZ
4230  015340  005114                  COM     (R4)            ;(R4)=177777
4231  015342  161314                  SUB     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=0100
4232  015344  103402                  BCS     SUB1
4233  015346  102401                  BVS     SUB1
4234  015350  001401                  BEQ     .+4
```

```
4235  015352  104400                SUB1:   HLT
4236
4237  015354  105013                        CLRB    (R3)            ; (R3)=177400
4238  015356  000313                        SWAB    (R3)            ;(R3)=000377
4239  015360  000270                        SEN
4240  015362  011314                        MOV     (R3),(R4)       ;(R3)=(R4)=000377
4241  015364  100001                        BPL     .+4
4242  015366  104400                        HLT
4243  015370  000314                        SWAB    (R4)            ;(R3)=000377,(R4)=177400
4244  015372  000263                        +SEC!SEV                ;SET C & V
4245  015374  051314                        BIS     (R3),(R4)       ;(R3)=000377,(R4)=177777, CC=1001
4246  015376  103002                        BCC     BIS1
4247  015400  102401                        BVS     BIS1
4248  015402  100401                        BMI     .+4
4249  015404  104400                BIS1:   HLT
4250
4251  015406  041314                        BIC     (R3),(R4)       ;(R3)=000377,(R4)=177400, CC=1001
4252  015410  103002                        BCC     BIC1
4253  015412  102401                        BVS     BIC1
4254  015414  100401                        BMI     .+4
4255  015416  104400                BIC1:   HLT
4256
4257  015420  000262                        SEV                     ;SET V
4258  015422  021314                        CMP     (R3),(R4)       ;(R3)=000377,(R4)=177400, CC=0001
4259  015424  103003                        BCC     CMP1A
4260  015426  102402                        BVS     CMP1A
4261  015430  001401                        BEQ     CMP1A
4262  015432  100001                        BPL     .+4
4263  015434  104400                CMP1A:  HLT
4264
4265  015436  005013                        CLR     (R3)            ;(R3)=000000
4266  015440  000261                        SEC
4267  015442  006013                        ROR     (R3)            ;(R3)=100000
4268  015444  011314                        MOV     (R3),(R4)       ;(R3)=(R4)=100000
4269  015446  005114                        COM     (R4)            ;(R4)=077777
4270  015450  161314                        SUB     (R3),(R4)       ;(R3)=100000,(R4)=177777, CC=1011
4271  015452  103002                        BCC     SUB1A
4272  015454  102001                        BVC     SUB1A
4273  015456  100401                        BMI     .+4
4274  015460  104400                SUB1A:  HLT
4275
4276  015462  000277                        SCC
4277  015464  161314                        SUB     (R3),(R4)       ;(R3)=100000,(R4)=077777, CC=0000
4278  015466  101402                        BLOS    SUB1B           ;BRANCH IF C OR Z IS SET
4279  015470  102401                        BVS     SUB1B
4280  015472  100001                        BPL     .+4
4281  015474  104400                SUB1B:  HLT
4282
4283  015476  011314                        MOV     (R3),(R4)       ;(R3)=100000,(R4)=100000, CC=1000
4284  015500  001401                        BEQ     MOV1
4285  015502  100401                        BMI     .+4
4286  015504  104400                MOV1:   HLT
4287
4288  015506  061314                        ADD     (R3),(R4)       ;(R3)=100000,(R4)=000000, CC=0111
4289  015510  103003                        BCC     ADD1A
4290  015512  102002                        BVC     ADD1A
```

```
4291  015514  001001                  BNE     ADD1A
4292  015516  100001                  BPL     .+4
4293  015520  104400          ADD1A:  HLT
4294
4295  015522  005113                  COM     (R3)            ;(R3)=077777
4296  015524  011314                  MOV     (R3),(R4)       ;(R4)=077777
4297  015526  061314                  ADD     (R3),(R4)       ;(R3)=077777,(R4)=177776, CC=1010
4298  015530  103402                  BCS     ADD1B
4299  015532  102001                  BVC     ADD1B
4300  015534  100401                  BMI     .+4
4301  015536  104400          ADD1B:  HLT
4302
4303  015540  062714  000002          ADD     #2,(R4)
4304  015544  005714                  TST     (R4)            ;CHECK FINAL RESULT
4305  015546  001401                  BEQ     .+4
4306  015550  104400                  HLT
4307  015552  104000                  SCOPE
4308
4309                          ;CHECK BINARY BYTE OPS USING ADDRESS MODE 1
4310  015554  000402                  BR      .+6
4311  015556  000000                  .WORD   0
4312  015560  000000                  .WORD   0
4313  015562  010705                  MOV     PC,R5
4314  015564  005745                  TST     -(R5)
4315  015566  005045                  CLR     -(R5)           ;(R5)=000000
4316  015570  010502                  MOV     R5,R2
4317  015572  005042                  CLR     -(R2)           ;(R2)=000000
4318  015574  005202                  INC     R2              ;R2 POINTS TO ODD BYTE
4319  015576  105112                  COMB    (R2)            ;(R2)=177400
4320
4321  015600  000277                  SCC
4322  015602  111215                  MOVB    (R2),(R5)       ;(R2)=177400,(R5)=000377,CC=1001
4323  015604  103005                  BCC     MOVB1
4324  015606  102404                  BVS     MOVB1
4325  015610  001403                  BEQ     MOVB1
4326  015612  100002                  BPL     MOVB1
4327  015614  105215                  INCB    (R5)            ;CHECK RESULT
4328  015616  001401                  BEQ     .+4
4329  015620  104400          MOVB1:  HLT
4330
4331  015622  106312                  ASLB    (R2)            ;SHIFT (R2) UNTIL
4332  015624  102376                  BVC     .-2             ;(R2)=000000
4333  015626  106012                  RORB    (R2)            ;(R2)=100000
4334  015630  105315                  DECB    (R5)            ;(R5)=00377
4335  015632  106015                  RORB    (R5)            ;(R5)=000177
4336  015634  000257                  CCC
4337  015636  121512                  CMPB    (R5),(R2)       ;(R5)=000177,(R2)=100000, CC=1010
4338  015640  102001                  BVC     CMPB1
4339  015642  100401                  BMI     .+4
4340  015644  104400          CMPB1:  HLT
4341
4342  015646  005003                  CLR     R3
4343  015650  000261                  SEC
4344  015652  006003                  ROR     R3              ;R3=100000
4345  015654  050315                  BIS     R3,(R5)         ;(R5)=100177
4346  015656  000273                  +SEC!SEV!SEN            ;SET C,V, & N
```

```
4347  015660  131215                BITB    (R2),(R5)       ;(R2)=100000,(R5)=100177, CC=0101
4348  015662  103002                BCC     BITB1
4349  015664  102401                BVS     BITB1
4350  015666  001401                BEQ     .+4
4351  015670  104400      BITB1:    HLT
4352
4353  015672  151215                BISB    (R2),(R5)       ;(R2)=100000,(R5)=100377, CC=1001
4354  015674  103001                BCC     BISB1
4355  015676  100401                BMI     .+4
4356  015700  104400      BISB1:    HLT
4357
4358  015702  141215                BICB    (R2),(R5)       ;(R2)=100000,(R5)=100177, CC=0001
4359  015704  103002                BCC     BICB1
4360  015706  001401                BEQ     BICB1
4361  015710  100001                BPL     .+4
4362  015712  104400      BICB1:    HLT
4363
4364  015714  105112                COMB    (R2)            ;(R2)=077400,(R5)=100177
4365  015716  121215                CMPB    (R2),(R5)
4366  015720  001401                BEQ     .+4
4367  015722  104400                HLT
4368
4369  015724  141512                BICB    (R5),(R2)       ;(R5)=100177,(R2)=000000, CC=0100
4370  015726  001002                BNE     BICB1A
4371  015730  105712                TSTB    (R2)
4372  015732  001401                BEQ     .+4
4373  015734  104400      BICB1A:   HLT
4374
4375  015736  000402                BR      .+6             ;RESERVE TWO WORDS FOR DATA
4376  015740  000000                .WORD   0               ;SOURCE DATA
4377  015742  000000                .WORD   0               ;DEST DATA
4378  015744  010705                MOV     PC,R5
4379  015746  005745                TST     -(R5)
4380  015750  105045                CLRB    -(R5)           ;R5 POINTS TO DEST ODD BYTE
4381  015752  010504                MOV     R5,R4
4382  015754  105044                CLRB    -(R4)           ;R4 POINTS TO DEST EVEN BYTE
4383  015756  010403                MOV     R4,R3
4384  015760  105043                CLRB    -(R3)           ;R3 POINTS TO SOURCE ODD BYTE
4385  015762  010302                MOV     R3,R2
4386  015764  105042                CLRB    -(R2)           ;R2 POINTS TO SOURCE EVEN BYTE
4387
4388                          ;COMMENTS ARE LEAST SIGNIFICANT 4 BITS OF BYTES POINTED TO BY R2,R3
4389                          ;R4, AND R5 RESPECTIVELY AND THE REMAINING BITS ARE 0'S.
4390  015766  000261                SEC                     ;SET CARRY
4391                                                        ;(R2),(R3),(R4),(R5)
4392  015770  106112                ROLB    (R2)            ;0001,0000,0000,0000
4393  015772  111214                MOVB    (R2),(R4)       ;0001,0000,0001,0000
4394  015774  106112                ROLB    (R2)            ;0010,0000,0001,0000
4395  015776  111213                MOVB    (R2),(R3)       ;0010,0010,0001,0000
4396  016000  106112                ROLB    (R2)            ;0100,0010,0001,0000
4397  016002  111315                MOVB    (R3),(R5)       ;0100,0010,0001,0010
4398  016004  106112                ROLB    (R2)            ;1000,0010,0001,0010
4399  016006  106113                ROLB    (R3)            ;1000,0100,0001,0010
4400  016010  151215                BISB    (R2),(R5)       ;1000,0100,0001,1010
4401  016012  131512                BITB    (R5),(R2)       ;1000,0100,0001,1010
4402  016014  001426                BEQ     BIN1
```

```
4403  016016  151314                  BISB    (R3),(R4)       ;1000,0100,0101,1010
4404  016020  131413                  BITB    (R4),(R3)       ;1000,0100,0101,1010
4405  016022  001423                  BEQ     BIN1
4406  016024  105213                  INCB    (R3)            ;1000,0101,0101,1010
4407  016026  121314                  CMPB    (R3),(R4)       ;1000,0101,0101,1010
4408  016030  001020                  BNE     BIN1
4409  016032  106113                  ROLB    (R3)            ;1000,1010,0101,1010
4410  016034  121315                  CMPB    (R3),(R5)       ;1000,1010,0101,1010
4411  016036  001015                  BNE     BIN1
4412  016040  106212                  ASRB    (R2)            ;0100,1010,0101,1010
4413  016042  131214                  BITB    (R2),(R4)       ;0100,1010,0101,1010
4414  016044  001412                  BEQ     BIN1
4415  016046  106015                  RORB    (R5)            ;0100,1010,0101,0101
4416  016050  121415                  CMPB    (R4),(R5)       ;0100,1010,0101,0101
4417  016052  001007                  BNE     BIN1
4418  016054  105314                  DECB    (R4)            ;0100,1010,0100,0101
4419  016056  141214                  BICB    (R2),(R4)       ;0100,1010,0000,0101
4420  016060  001004                  BNE     BIN1
4421  016062  111314                  MOVB    (R3),(R4)       ;0100,1010,1010,0101
4422  016064  106213                  ASRB    (R3)            ;0100,0101,1010,0101
4423  016066  141315                  BICB    (R3),(R5)       ;0100,0101,1010,0101
4424  016070  001401                  BEQ     .+4
4425  016072  104400          BIN1:   HLT
4426  016074  104000                  SCOPE
4427
4428                          ;CHECK BINARY WORD OPS USING ADDRESS MODE 2 & 4
4429  016076  010405                  MOV     R4,R5           ;SET DESTINATION REGISTER
4430  016100  012715  000001          MOV     #1,(R5)
4431  016104  012712  177777          MOV     #-1,(R2)
4432  016110  000257                  CCC
4433  016112  000262                  SEV
4434  016114  062225                  ADD     (R2)+,(R5)+     ;(R2)=177777,(R5)=000000, CC=0101
4435  016116  103002                  BCC     ADD2
4436  016120  102401                  BVS     ADD2
4437  016122  001401                  BEQ     .+4
4438  016124  104400          ADD2:   HLT
4439
4440  016126  000262                  SEV                     ;SET-V
4441  016130  024527  000001          CMP     -(R5),#1        ;(R5)=000000, CC=1001
4442  016134  103002                  BCC     CMP2
4443  016136  102401                  BVS     CMP2
4444  016140  100401                  BMI     .+4
4445  016142  104400          CMP2:   HLT
4446
4447  016144  054225                  BIS     -(R2),(R5)+     ;(R2)=177777,(R5)=177777,CC=1001
4448  016146  103001                  BCC     BIS2
4449  016150  100401                  BMI     .+4
4450  016152  104400          BIS2:   HLT
4451  016154  000277                  SCC
4452  016156  000244                  CLZ
4453  016160  162245                  SUB     (R2)+,-(R5)     ;(R2)=177777,(R5)=000000, CC=0100
4454  016162  103402                  BCS     SUB2
4455  016164  102401                  BVS     SUB2
4456  016166  001401                  BEQ     .+4
4457  016170  104400          SUB2:   HLT
4458
```

```
4459  016172  005442                        NEG     -(R2)           ;(R2)=000001
4460  016174  005115                        COM     (R5)            ;(R5)=177777
4461  016176  000277                        SCC
4462  016200  000250                        CLN
4463  016202  042225                        BIC     (R2)+,(R5)+     ;(R2)=000001,(R5)=177776, CC=1001
4464  016204  103003                        BCC     BIC2
4465  016206  102402                        BVS     BIC2
4466  016210  001401                        BEQ     BIC2
4467  016212  100401                        BMI     .+4
4468  016214  104400              BIC2:     HLT
4469
4470  016216  012742  125252                MOV     #125252,-(R2)
4471  016222  012245                        MOV     (R2)+,-(R5)
4472  016224  005125                        COM     (R5)+           ;(R5)=052525
4473  016226  000262                        SEV
4474  016230  034245                        BIT     -(R2),-(R5)     ;(R2)=125252,(R5)=052525, CC=0101
4475  016232  103002                        BCC     BIT2
4476  016234  102401                        BVS     BIT2
4477  016236  001401                        BEQ     .+4
4478  016240  104400              BIT2:     HLT
4479
4480  016242  000262                        SEV
4481  016244  052225                        BIS     (R2)+,(R5)+     ;(R2)=125252,(R5)=177777, CC=1001
4482  016246  103002                        BCC     BIS2A
4483  016250  102401                        BVS     BIS2A
4484  016252  100401                        BMI     .+4
4485  016254  104400              BIS2A:    HLT
4486
4487  016256  042745  125252                BIC     #125252,-(R5)   ;(R5)=052525
4488  016262  005125                        COM     (R5)+           ;(R5)=125252
4489  016264  024245                        CMP     -(R2),-(R5)
4490  016266  001401                        BEQ     .+4
4491  016270  104400                        HLT
4492
4493  016272  005012                        CLR     (R2)
4494  016274  005122                        COM     (R2)+           ;(R2)=177777
4495  016276  162742  000001                SUB     #1,-(R2)        ;(R2)=177776, CC=1000
4496  016302  103402                        BCS     SUB2A
4497  016304  102401                        BVS     SUB2A
4498  016306  100401                        BMI     .+4
4499  016310  104400              SUB2A:    HLT
4500  016312  104000                        SCOPE
4501
4502  016314  010702                        MOV     PC,R2           ;GET CURRENT PC
4503  016316  010205                        MOV     R2,R5           ;MOVE TO R5
4504  016320  124245              1$:       CMPB    -(R2),-(R5)     ;COMPARE ALL PREVIOUS MEMORY ADDRESSES
4505  016322  001401                        BEQ     .+4
4506  016324  104400                        HLT                     ;ERROR!
4507  016326  020237  001010                CMP     R2,@#FRSTAD     ;CHECK FOR LOW LIMIT
4508  016332  001372                        BNE     1$
4509  016334  104000                      .SCOPE
4510
4511                                  ;CHECK BINARY BYTE OPS USING ADDRESS MODES 2 & 4.
4512  016336  000402                        BR      .+6             ;RESERVE TWO WORDS
4513  016340  000000                        .WORD   0               ;SOURCE DATA
4514  016342  000000                        .WORD   0               ;DESTINATION DATA
```

```
4515  016344  010703                  MOV     PC,R3
4516  016346  005743                  TST     -(R3)
4517
4518                                  ;FIRST CHECK AUTO INCREMENT/DECREMENT
4519  016350  010300                  MOV     R3,R0
4520  016352  010002                  MOV     R0,R2
4521  016354  005302                  DEC     R2
4522  016356  010604                  MOV     SP,R4
4523  016360  010605                  MOV     SP,R5
4524  016362  005745                  TST     -(R5)
4525
4526  016364  114046                  MOVB    -(R0),-(SP)
4527  016366  020506                  CMP     R5,SP
4528  016370  001021                  BNE     BINB
4529  016372  020200                  CMP     R2,R0
4530  016374  001017                  BNE     BINB
4531  016376  122026                  CMPB    (R0)+,(SP)+
4532  016400  020406                  CMP     R4,SP
4533  016402  001014                  BNE     BINB
4534  016404  020003                  CMP     R0,R3
4535  016406  001012                  BNE     BINB
4536  016410  154640                  BISB    -(SP),-(R0)
4537  016412  020506                  CMP     R5,SP
4538  016414  001007                  BNE     BINB
4539  016416  020200                  CMP     R2,R0
4540  016420  001005                  BNE     BINB
4541  016422  142620                  BICB    (SP)+,(R0)+
4542  016424  020406                  CMP     R4,SP
4543  016426  001002                  BNE     BINB
4544  016430  020003                  CMP     R0,R3
4545  016432  001401                  BEQ     .+4
4546  016434  104400          BINB:   HLT
4547  016436  104000                  SCOPE
4548
4549  016440  010003                  MOV     R0,R3
4550  016442  112743  000200          MOVB    #200,-(R3)
4551  016446  112743  000377          MOVB    #377,-(R3)      ;(R3)=100377
4552  016452  010304                  MOV     R3,R4
4553  016454  112744  000177          MOVB    #177,-(R4)
4554  016460  112744  000000          MOVB    #0,-(R4)        ;(R4)=077400
4555  016464  001401                  BEQ     .+4
4556  016466  104400                  HLT
4557
4558  016470  152324                  BISB    (R3)+,(R4)+     ;(R3)=100377,(R4)=077777
4559  016472  100401                  BMI     .+4
4560  016474  104400                  HLT
4561
4562  016476  122324                  CMPB    (R3)+,(R4)+
4563  016500  103402                  BCS     CMPB2
4564  016502  102001                  BVC     CMPB2
4565  016504  100001                  BPL     .+4
4566  016506  104400          CMPB2:  HLT
4567
4568  016510  000261                  SEC
4569  016512  134344                  BITB    -(R3),-(R4)
4570  016514  103002                  BCC     BITB2
```

```
 4571  016516  102401                BVS     BITB2
 4572  016520  001401                BEQ     .+4
 4573  016522  104400          BITB2: HLT
 4574
 4575  016524  000244                CLZ
 4576  016526  144344                BICB    -(R3),-(R4)     ;(R3)=100377,(R4)=077400
 4577  016530  001401                BEQ     .+4
 4578  016532  104400                HLT
 4579  016534  104000                SCOPE
 4580
 4581                          ;CHECK BINARY WORD OPS USING ADDRESS MODES 3 & 5.
 4582  016536  000404                BR      2$              ;RESERVE SPACE FOR DATA AND ADDRESSES
 4583  016540  000000                .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA
 4584  016542  000000                .WORD   0               ;CONTAINS ADDRESS OF DEST DATA
 4585  016544  000000                .WORD   0               ;CONTAINS SOURCE DATA
 4586  016546  000000                .WORD   0               ;CONTAINS DEST DATA
 4587  016550  010701          2$:   MOV     PC,R1
 4588  016552  010100                MOV     R1,R0           ;SET SCOPE PTR
 4589  016554  024040                CMP     -(R0),-(R0)     ;ADJUST R0
 4590  016556  010005                MOV     R0,R5           ;R5 POINTS TO DEST DATA
 4591  016560  024545                CMP     -(R5),-(R5)     ;SUB 4 FROM R5
 4592  016562  010015                MOV     R0,(R5)         ;R5 POINTS TO ADDRESS OF DEST DATA
 4593  016564  010502                MOV     R5,R2
 4594  016566  010004                MOV     R0,R4           ;R4 POINTS TO DEST DATA
 4595  016570  005740                TST     -(R0)
 4596  016572  010003                MOV     R0,R3           ;R3 POINTS TO SOURCE DATA
 4597  016574  010042                MOV     R0,-(R2)        ;R2 POINTS TO ADDRESS OF SOURCE DATA
 4598  016576  005013                CLR     (R3)            ;PRESET SOURCE DATA
 4599  016600  005014                CLR     (R4)            ;PRESET DEST DATA
 4600
 4601  016602  000277                SCC
 4602  016604  000244                CLZ
 4603  016606  163235                SUB     @(R2)+,@(R5)+   ;(R3)=000000,(R4)=000000, CC=0100
 4604  016610  103402                BCS     SUB3
 4605  016612  102401                BVS     SUB3
 4606  016614  001401                BEQ     .+4
 4607  016616  104400          SUB3: HLT
 4608
 4609  016620  052752  100000        BIS     #100000,@-(R2)  ;(R3)=100000
 4610  016624  062755  000001        ADD     #1,@-(R5)       ;(R4)=000001
 4611  016630  163235                SUB     @(R2)+,@(R5)+   ;(R3)=100000,(R4)=100001, CC=1011
 4612  016632  103002                BCC     SUB3A
 4613  016634  102001                BVC     SUB3A
 4614  016636  100401                BMI     .+4
 4615  016640  104400          SUB3A: HLT
 4616
 4617  016642  005414                NEG     (R4)            ;(R4)=077777
 4618  016644  035255                BIT     @-(R2),@-(R5)   ;(R3)=100000,(R4)=077777
 4619  016646  001401                BEQ     .+4
 4620  016650  104400                HLT
 4621  016652  023235                CMP     @(R2)+,@(R5)+
 4622  016654  102401                BVS     .+4
 4623  016656  104400                HLT
 4624  016660  005152                COM     @-(R2)
 4625  016662  000257                CCC
 4626  016664  063255                ADD     @(R2)+,@-(R5)
```

```
4627  016666  102001                  BVC     ADD3
4628  016670  100401                  BMI     .+4
4629  016672  104400          ADD3:   HLT
4630  016674  000261                  SEC
4631  016676  045235                  BIC     @-(R2),@(R5)+   ;(R3)=077777,(R4)=100000
4632  016700  103001                  BCC     BIC3
4633  016702  100401                  BMI     .+4
4634  016704  104400          BIC3:   HLT
4635
4636  016706  005155                  COM     @-(R5)          ;(R4)=077777
4637  016710  023235                  CMP     @(R2)+,@(R5)+   ;(R3)=077777,(R4)=077777
4638  016712  001401                  BEQ     .+4
4639  016714  104400                  HLT
4640  016716  104000                  SCOPE
4641
4642                          ;CHECK BINARY BYTE OPS USING ADDRESS MODES 3 & 5.
4643  016720  000406                  BR      1$              ;RESERVE SPACE FOR ADDRESSES & DATA
4644  016722  000000                  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA (EVEN BYTE)
4645  016724  000000                  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA (ODD BYTE)
4646  016726  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA (EVEN BYTE)
4647  016730  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA (ODD BYTE)
4648  016732  000000                  .WORD   0               ;CONTAINS SOURCE DATA
4649  016734  000000                  .WORD   0               ;CONTAINS DEST DATA
4650
4651  016736  010700          1$:     MOV     PC,R0
4652  016740  024040                  CMP     -(R0),-(R0)     ;R0=ADDRESS OF DEST DATA
4653  016742  010003                  MOV     R0,R3           ;R3     "       "
4654  016744  010305                  MOV     R3,R5           ;R5     "       "
4655  016746  005743                  TST     -(R3)           ;SUB 2 FROM R3
4656  016750  010043                  MOV     R0,-(R3)        ;R3 POINTS TO ADDRESS OF DEST DATA
4657  016752  005213                  INC     (R3)            ;ODD BYTE
4658  016754  010043                  MOV     R0,-(R3)        ;EVEN BYTE
4659  016756  010304                  MOV     R3,R4
4660  016760  005740                  TST     -(R0)           ;R0=ADDRESS OF SOURCE DATA
4661  016762  010044                  MOV     R0,-(R4)        ;R4 POINTS TO ADDRESS OF SOURCE DATA
4662  016764  005214                  INC     (R4)            ;ODD BYTE
4663  016766  010044                  MOV     R0,-(R4)        ;EVEN BYTE
4664
4665  016770  000261                  SEC                     ;SET CARRY
4666  016772  012734  177001          MOV     #177001,@(R4)+
4667  016776  112734  000200          MOVB    #200,@(R4)+     ;SOURCE DATA=100001
4668  017002  115433                  MOVB    @-(R4),@(R3)+
4669  017004  115433                  MOVB    @-(R4),@(R3)+   ;DEST DATA=000600
4670  017006  103401                  BCS     .+4
4671  017010  104400                  HLT                     ;ERROR! MOV DOES AFFECT C BIT IN PSW
4672  017012  022715  000600          CMP     #600,(R5)       ;CHECK DEST DATA
4673  017016  001401                  BEQ     .+4
4674  017020  104400                  HLT                     ;ERROR! INCORRECT RESULT
4675  017022  024343                  CMP     -(R3),-(R3)     ;POINT R4 BACK TO EVEN BYTE
4676  017024  153433                  BISB    @(R4)+,@(R3)+
4677  017026  153433                  BISB    @(R4)+,@(R3)+   ;DEST DATA=100601
4678  017030  022715  100601          CMP     #100601,(R5)    ;CHECK RESULT
4679  017034  001401                  BEQ     .+4
4680  017036  104400                  HLT                     ;ERROR! INCORRECT DEST DATA AFTER BISB
4681  017040  145453                  BICB    @-(R4),@-(R3)
4682  017042  145453                  BICB    @-(R4),@-(R3)
```

```
 4683  017044  133433                    BITB    @(R4)+,@(R3)+
 4684  017046  001002                    BNE     BITB3
 4685  017050  135433                    BITB    @-(R4),@(R3)+
 4686  017052  001001                    BNE     .+4
 4687  017054  104400            BITB3:  HLT
 4688
 4689  017056  123453                    CMPB    @(R4)+,@-(R3)
 4690  017060  001002                    BNE     CMPB3
```

```
4691  017062  123453                          CMPB    @(R4)+,@-(R3)
4692  017064  001401                          BEQ     .+4
4693  017066  104400              CMPB3:      HLT
4694  017070  104000                          SCOPE
4695
4696                              ;CHECK BINARY OPS USING ADDRESS MODE 6
4697  017072  000402                          BR      .+6             ;RESERVE TWO LOCATIONS
4698  017074  000000              SDATA:      .WORD   0               ;RESERVED FOR SOURCE DATA
4699  017076  000000              DDATA:      .WORD   0               ;RESERVED FOR DESTINATION DATA
4700
4701  017100  013702  000004                  MOV     @#FACTOR,R2     ;GET RELOCATION FACTOR AND USE AS AN
4702  017104  010205                          MOV     R2,R5           ;INDEX VALUE TO POINT TO DATA
4703  017106  005065  017076                  CLR     DDATA(5)        ;PRESET DESTINATION DATA
4704  017112  012762  000001  017074          MOV     #1,SDATA(2)     ;THIS ROUTSINE PUT A 1 BIT INTO EVERY
4705  017120  056265  017074  017076  1$:     BIS     SDATA(2),DDATA(5)       ;OTHER BIT POSITION IN THE DEST-
4706  017126  006362  017074                  ASL     SDATA(2)                ;INATION ADDRESS (52525)
4707  017132  006362  017074                  ASL     SDATA(2)
4708  017136  103370                          BCC     1$
4709  017140  022765  052525  017076          CMP     #52525,DDATA(5) ;CHECK RESULT
4710  017146  001401                          BEQ     .+4
4711  017150  104400                          HLT                     ;ERROR! INCORRECT RESULT
4712  017154  012762  177777  017074          MOV     #-1,SDATA(2)
4713  017160  046562  017076  017074          BIC     DDATA(5),SDATA(2)       ;SOURCE DATA=125252
4714  017166  036265  017074  017076          BIT     SDATA(2),DDATA(5)
4715  017174  001401                          BEQ     .+4
4716  017176  104400                          HLT                     ;ERROR! BIT INST FAILED
4717  017200  006365  017076                  ASL     DDATA(5)        ;DDATA=125252
4718  017204  026265  017074  017076          CMP     SDATA(2),DDATA(5)
4719  017212  001401                          BEQ     .+4
4720
4721  017214  104400                          HLT                     ;ERROR! CMP INST FAILED
4722  017216  000257                          CCC
4723  017220  066265  017074  017076          ADD     SDATA(2),DDATA(5)
4724  017226  103002                          BCC     ADD6
4725  017230  102001                          BVC     ADD6
4726  017232  100001                          BPL     .+4
4727  017234  104400              ADD6:       HLT
4728
4729  017236  006362  017074                  ASL     SDATA(2)        ;SDATA=52524
4730  017242  166265  017074  017076          SUB     SDATA(2),DDATA(5)
4731  017250  103401                          BCS     SUB6
4732  017252  001401                          BEQ     .+4
4733  017254  104400              SUB6:       HLT
4734
4735  017256  112700  000377                  MOVB    #377,R0         ;R0=177777 (MOVB %R EXTENDS SIGN)
4736  017262  010062  017074                  MOV     R0,SDATA(2)
4737  017266  012765  177777  017076          MOV     #-1,DDATA(5)
4738  017274  166500  017076                  SUB     DDATA(5),R0
4739  017300  001401                          BEQ     .+4
4740  017302  104400                          HLT
4741  017304  066265  017074  017076  1$:     ADD     SDATA(2),DDATA(5)
4742  017312  006362  017074                  ASL     SDATA(2)
4743  017316  005162  017074                  COM     SDATA(2)
4744  017322  036265  017074  017076          BIT     SDATA(2),DDATA(5)
4745  017330  001401                          BEQ     .+4
4746  017332  104400                          HLT
```

```
4747  017334  005162  017074                  COM     SDATA(2)
4748  017340  026265  017074  017076          CMP     SDATA(2),DDATA(5)
4749  017346  001401                          BEQ     .+4
4750  017350  104400                          HLT
4751  017352  026200  017074                  CMP     SDATA(2),R0
4752  017356  001352                          BNE     1$
4753  017360  104000                          SCOPE
4754
4755                                  ;CHECK BINARY BYTE OPS USING ADDRESS MODE 6
4756                                  ;NOTE: SDATAB(2), AND DDATAB(4) REFERENCE EVEN BYTE OF SOURCE & DEST DATA
4757                                  ;AND SDATAB(3), AND DDATAB(5) REFERENCE ODD BYTE OF SOURCE & DEST DATA
4758
4759  017362  013702  001004                  MOV     @#FACTOR,R2     ;GET INDEX VALUE
4760  017366  010204                          MOV     R2,R4           ;R2 FOR SOURCE EVEN BYTE INDEX, R4 FOR
4761  017370  010403                          MOV     R4,R3           ;DEST ODD BYTE, R3 FOR SOURCE EVEN
4762  017372  005203                          INC     R3              ;AND R5 FOR DEST ODD BYTE
4763  017374  010305                          MOV     R3,R5
4764  017376  000261                          SEC                             ;SET CARRY
4765  017400  012762  125252  017524          MOV     #125252,SDATAB(2)
4766  017406  112763  177125  017524          MOVB    #177125,SDATAB(3)       ;SOURCE DATA = 052652
4767  017414  016264  017524  017526          MOV     SDATAB(2),DDATAB(4)
4768  017422  052764  125125  017526          BIS     #125125,DDATAB(4)       ;DEST DATA = 177777
4769  017430  136263  017524  017524          BITB    SDATAB(2),SDATAB(3)
4770  017436  001401                          BEQ     .+4
4771  017440  104400                  BITB6:  HLT
4772
4773  017442  146264  017524  017526          BICB    SDATAB(2),DDATAB(4)
4774  017450  103401                          BCS     .+4
4775  017452  104400                          HLT                     ;ERROR MOV,BIS,BIT;BIC DO NOT AFFECT 'C'
4776  017454  126364  017524  017526          CMPB    SDATAB(3),DDATAB(4)
4777  017462  001401                          BEQ     .+4
4778  017464  104400                          HLT
4779
4780  017466  146365  017524  017526          BICB    SDATAB(3),DDATAB(5)
4781  017474  126265  017524  017526          CMPB    SDATAB(2),DDATAB(5)
4782  017502  001401                          BEQ     .+4
4783  017504  104400                          HLT
4784
4785  017506  136564  017526  017526          BITB    DDATAB(5),DDATAB(4)
4786  017514  001401                          BEQ     .+4
4787  017516  104400                          HLT
4788  017520  104000                          SCOPE
4789
4790  017522  000406                          BR      UB7             ;RESERVE TWO WORDS
4791  017524  000000                  SDATAB: .WORD   0               ;RESERVED FOR SOURCE DATA
4792  017526  000000                  DDATAB: .WORD   0               ;RESERVED FOR DEST DATA
4793
4794                                  ;CHECK BINARY WORD OPS USING ADDRESS MODE 7
4795                                  ;R2=ADDRESS OF SOURCE DATA, AND R3= ADDRESS OF DEST DATA
4796  017530  000000                  SBIN7:  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA
4797  017532  000000                  DBIN7:  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA
4798  017534  000000                          .WORD   0               ;CONTAINS SOURCE DATA
4799  017536  000000                          .WORD   0               ;CONTAINS DEST DATA
4800
4801  017540  010700                  UB7:    MOV     PC,R0
4802  017542  024040                          CMP     -(R0),-(R0)
```

```
4803  017544  010002                        MOV     R0,R2
4804  017546  024242                        CMP     -(R2),-(R2)
4805  017550  010012                        MOV     R0,(R2)
4806  017552  010203                        MOV     R2,R3
4807  017554  024043                        CMP     -(R0),-(R3)
4808  017556  010013                        MOV     R0,(R3)
4809
4810  017560  000261                        SEC
4811  017562  012777  100000  177740        MOV     #100000,@SBIN7  ;SOURCE DATA = 100000
4812  017570  017777  177734  177734        MOV     @SBIN7,@DBIN7   ;DEST DATA = 100000
4813  017576  103001                        BCC     MOV7
4814  017600  100401                        BMI     .+4
4815  017602  104400                MOV7:   HLT
4816  017604  006377  177722                ASL     @DBIN7          ;DEST DATA = 000000
4817  017610  102001                        BVC     .+4
4818  017612  001401                        BEQ     .+4
4819  017614  104400                        HLT
4820
4821  017616  027777  177706  177706        CMP     @SBIN7,@DBIN7   ;(R2)=100000,(R3)=000000
4822  017624  103402                        BCS     CMP7
4823  017626  102401                        BVS     CMP7
4824  017630  100401                        BMI     .+4
4825  017632  104400                CMP7:   HLT
4826
4827  017634  167777  177670  177670        SUB     @SBIN7,@DBIN7   ;(R2)=100000,(R3)=100000
4828  017642  103003                        BCC     SUB7
4829  017644  102002                        BVC     SUB7
4830  017646  001401                        BEQ     SUB7
4831  017650  100401                        BMI     .+4
4832  017652  104400                SUB7:   HLT
4833
4834  017654  006277  177650                ASR     @SBIN7          ;(R2)=140000
4835  017660  067777  177644  177644        ADD     @SBIN7,@DBIN7   ;(R2)=140000,(R3)=040000
4836  017666  103003                        BCC     ADD7
4837  017670  102002                        BVC     ADD7
4838  017672  001401                        BEQ     ADD7
4839  017674  100001                        BPL     .+4
4840  017676  104400                ADD7:   HLT
4841
4842  017700  047777  177624  177624        BIC     @SBIN7,@DBIN7   ;(R2)=140000,(R3)=000000
4843  017706  001401                        BEQ     .+4
4844  017710  104400                        HLT
4845
4846  017712  057777  177612  177612        BIS     @SBIN7,@DBIN7   ;(R2)=140000,(R3)=140000
4847  017720  100401                        BMI     .+4
4848  017722  104400                        HLT
4849
4850  017724  027777  177600  177600        CMP     @SBIN7,@DBIN7
4851  017732  001401                        BEQ     .+4
4852  017734  104400                        HLT
4853  017736  104000                        SCOPE
4854
4855                                ;SOME MISCELLANEOUS OPERATION INVOLVING THE PC
4856                                ;NOTE: NONE OF THESE OPERATIONS SHOULD AFFECT THE PC
4857  017740  005000                        CLR     R0
4858  017742  005067  000072                CLR     1$
```

```
4853  017746  010707                          MOV     PC,PC
4860  017750  120707                          CMPB    PC,PC
486.  017752  030707                          BIT     PC,PC
4862  017754  060007                          ADD     R0,PC
486.  017756  105707                          TSTB    PC
4864  017760  005507                          ADC     PC
4865  017762  021007                          CMP     (R0),PC
4866  017764  131007                          BITB    (R0),PC
4867  017766  062707  000000                  ADD     #0,PC
4868  017772  023707  001004                  CMP     @#FACTOR,PC
4869  017776  133707  001004                  BITB    @#FACTOR,PC
4870  020002  000240                          NOP
4871                                  ;THE NEXT TWO INSTRUCTION CAUSE THE PROGRAM TO JUMP TO THE UNRELOCATED
4872                                  ;CODE AND TO RETURN ON THE FOLLOWING INST (IF THE CODE IS RELOCATED)
4873  020004  163707  001004                  SUB     @#FACTOR,PC     ;JUMPS TO UNRELOCATED CODE
4874  020010  063707  001004                  ADD     @#FACTOR,PC     ;RETURNS
4875  020014  000240                          NOP
4876  020016  024607                          CMP     -(SP),PC
4877  020020  132607                          BITB    (SP)+,PC
4878  020022  026707  000012                  CMP     1$,PC
4879  020026  166707  000006                  SUB     1$,PC
4880  020032  046707  000002                  BIC     1$,PC
4881  020036  000401                          BR      .+4             ;BRANCH OVER 1$
4882  020040  000000                  1$:     0
4883  020042  104000                          SCOPE
4884
4885  020044  010702                          MOV     PC,R2
4886  020046  062702  000012                  ADD     #12,R2
4887  020052  012707  001152                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
4888  020056  000000                  REL22:  .WORD   0
4889                                  ;2222222222222 LAST ADDRESS OF CODE TO BE RELOCATED 22222222222
4890
4891
4892                                          .SBTTL  START OF SECTION 3
4893                                  ;3333333333333 FIRST ADDRESS TO BE RELOCATED 333333333
4894  020060  010700                  REL3:   MOV     PC,R0           ;GET PC
4895  020062  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL3
4896  020064  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
4897  020070  012737  000003  005442          MOV     #3,@#SECT       ;SET SECTION #
4898  020076  004737  005432                  JSR     PC,@#LDDISP     ;LOAD DISPLAY GEG
4899  020102  013767  005436  002104          MOV     @#DISPLY,REL33
4900  020110  010700                          MOV     PC,R0           ;GET CURRENT PC
4901  020112  162700  020112                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
4902  020116  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
4903  020122  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR
4904
4905                                  ;CHECK BINARY BYTE OPS USING ADDRESS MODE 0
4906  020124  012703  125252                  MOV     #125252,R3
4907  020130  010304                          MOV     R3,R4           ;R3=R4=125252
4908  020132  140304                          BICB    R3,R4           ;R3=125252,R4=125000
4909  020134  022704  125000                  CMP     #125000,R4      ;CHECK RESULT
4910  020140  001401                          BEQ     .+4
4911  020142  104400                          HLT
4912
4913  020144  005004                          CLR     R4              ;R3=125252,R4=0
4914  020146  150304                          BISB    R3,R4           ;R3=125252,R4=000252
```

```
 4915  020150  022704  000252          CMP     #252,R4         ;CHECK RESULT
 4916  020154  001401                  BEQ     .+4
 4917  020156  104400                  HLT
 4918
 4919  020160  110404                  MOVB    R4,R4           ;R4=177652
 4920  020162  022704  177652          CMP     #177652,R4      ;CHECK RESULT
 4921  020166  001401                  BEQ     .+4
 4922  020170  104400                  HLT
 4923
 4924  020172  132704  177525          BITB    #177525,R4
 4925  020176  001401                  BEQ     .+4
 4926  020200  104400                  HLT
 4927
 4928  020202  105104                  COMB    R4              ;R4=177525
 4929  020204  110404                  MOVB    R4,R4           ;R4=000125
 4930  020206  022704  000125          CMP     #125,R4         ;CHECK RESULT
 4931  020212  001401                  BEQ     .+4
 4932  020214  104400                  HLT
 4933
 4934  020216  150304                  BISB    R3,R4           ;R3=125252,R4=000377
 4935  020220  105204                  INCB    R4
 4936  020222  001401                  BEQ     .+4
 4937  020224  104400                  HLT
 4938  020226  104000                  SCOPE
 4939
 4940                          ;CHECK BINARY BYTE OPS USING ADDRESS MODE 7
 4941  020230  000406                  BR      BINB7           ;RESERVE SPACE FOR ADDRESSES & DATA
 4942  020232  000000          SBINB7: .WORD   0               ;CONTAINS ADDRESS OF SOURCE EVEN BYTE
 4943  020234  000000                  .WORD   0               ;CONTAINS ADDRESS OF SOURCE ODD BYTE
 4944  020236  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST EVEN BYTE
 4945  020240  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST ODD BYTE
 4946  020242  000000          DBINB7: .WORD   0               ;CONTAINS SOURCE DATA
 4947  020244  000000                  .WORD   0               ;CONTAINS DEST DATA
 4948
 4949  020246  010700          BINB7:  MOV     PC,R0
 4950  020250  024040                  CMP     -(R0),-(R0)     ;R0 = ADDRESS OF DEST DATA
 4951  020252  010060  177772          MOV     R0,-6(R0)       ;LOAD ADDRES OF DEST EVEN BYTE DATA
 4952  020256  010060  177774          MOV     R0,-4(R0)
 4953  020262  005260  177774          INC     -4(R0)          ;LOAD ADDRESS OF DEST ODD BYTE DATA
```

```
 4954  020266  005740                          TST     -(R0)              ;R0=ADDRESS OF SOURCE DATA
 4955  020270  010060  177770                  MOV     R0,-10(R0)         ;LOAD ADDRESS OF SOURCE EVEN BYTE DATA
 4956  020274  010060  177772                  MOV     R0,-6(R0)
 4957  020300  005260  177772                  INC     -6(R0)             ;LOAD ADDRESS OF SOURCE ODD BYTE DATA
 4958
 4959  020304  005002                          CLR     R2                 ;SET INDEX REGISTERS
 4960  020306  012703  000002                  MOV     #2,R3              ;@SBINB7(2);@SBINB7(3) REFERENCE EVEN &
 4961  020312  012704  177774                  MOV     #-4,R4             ;ODD BYTE SOURCE DATA; @DBINB7(4);@DBINB7(5)
 4962  020316  012705  177776                  MOV     #-2,R5             ;REFERENCE DEST EVEN& ODD BYTE DATA
 4963
 4964
 4965  020322  005020                          CLR     (R0)+              ;PRESET SOURCE DATA
 4966  020324  005010                          CLR     (R0)               ;PRESET DEST DATA
 4967  020326  013746  001004                  MOV     @#FACTOR,-(SP)     ;GET RELOCATION FACTOR
 4968  020332  061602                          ADD     (SP),R2            ;AND ADD TO INDEX VALUES
 4969  020334  061603                          ADD     (SP),R3
 4970  020336  061604                          ADD     (SP),R4
 4971  020340  062605                          ADD     (SP)+,R5
 4972
 4973  020342  112773  177777  020232          MOVB    #-1,@SBINB7(3)     ;SRC DATA = 177400
 4974  020350  132772  000377  020232          BITB    #377,@SBINB7(2)    ;CHECK THAT EVEN BYTE WAS NOT AFFECTED
 4975  020356  001401                          BEQ     .+4                ;BY MOVB INSTRUCTION
 4976  020360  104400                          HLT
 4977
 4978  020362  157374  020232  020242          BISB    @SBINB7(3),@DBINB7(4)
 4979  020370  105274  020242                  INCB    @DBINB7(4)         ;CHECK THAT BIS SET ALL BITS
 4980  020374  001401                          BEQ     .+4
 4981  020376  104400                          HLT
 4982
 4983  020400  105375  020242                  DECB    @DBINB7(5)         ;DEST DATA = 177400
 4984  020404  005274  020242                  INC     @DBINB7(4)         ;DEST DATA = 177401
 4985  020410  127375  020232  020242          CMPB    @SBINB7(3),@DBINB7(5)
 4986  020416  001401                          BEQ     .+4
 4987  020420  104400                          HLT
 4988
 4989  020422  147375  020232  020242          BICB    @SBINB7(3),@DBINB7(5)
 4990  020430  001401                          BEQ     .+4
 4991  020432  104400                          HLT
 4992
 4993  020434  105073  020232                  CLRB    @SBINB7(3)         ;SRC DATA = 000000
 4994                                ;THIS ROUTINE SETS ALL BITS IN THE SOURCE ODD BYTE BY BISING A BIT FROM
 4995                                ;THE DEST EVEN BYTE INTO THE SOURCE ODD BYTE
 4996  020440  157473  020242  020232  BIS7:   BISB    @DBINB7(4),@SBINB7(3)
 4997  020446  106174  020242                  ROLB    @DBINB7(4)
 4998  020452  103372                          BCC     BIS7
 4999  020454  022772  177400  020232          CMP     #177400,@SBINB7(2)     ;CHECK RESULT
 5000  020462  001401                          BEQ     .+4
 5001  020464  104400                          HLT
 5002
 5003  020466  000372  020232                  SWAB    @SBINB7(2)         ;SRC DATA = 000377
 5004  020472  112775  000200  020242          MOVB    #200,@DBINB7(5)    ;DEST DATA = 100000
 5005
 5006  020500  147572  020242  020232  BIC7:   BICB    @DBINB7(5),@SBINB7(2)
 5007  020506  106075  020242                  RORB    @DBINB7(5)
 5008  020512  103372                          BCC     BIC7
 5009  020514  005772  020232                  TST     @SBINB7(2)
```

```
5010  020520  001401                        BEQ     .+4
5011  020522  104400                        HLT
5012  020524  104000                        SCOPE
5013
5014  020526  012702  000001        0AERR:  MOV     #1,R2           ;LOAD R2 WITH ODD #
5015  020532  010703                        MOV     PC,R3
5016  020534  000401                        BR      .+4             ;RESERVE SPACE FOR A WORD
5017  020536  000000                        .WORD   0               ;WILL CONTAIN AN ODD ADDRESS
5018  020540  005723                        TST     (R3)+           ;STEP R3 TO POINT TO WORD ABOVE
5019  020542  010313                        MOV     R3,(R3)
5020  020544  005213                        INC     (R3)            ;AND MAKE ODD
5021  020546  012737  020674  000004        MOV     #1$,@#ERRVEC    ;SET ODD ADDRESS & RESERVED INSTRUCTION
5022  020554  063737  001004  000004        ADD     @#FACTOR,@#ERRVEC
5023  020562  013737  000004  000010        MOV     @#ERRVEC,@#RESVEC       ;TO TRAP TO 1$ BELOW
5024
5025  020570  000277                        SCC                     ;SET ALL CC'S
5026  020572  160212                        SUB     R2,(R2)
5027  020574  104400                        HLT
5028  020576  060222                        ADD     R2,(R2)+
5029  020600  104400                        HLT
5030  020602  006342                        ASL     -(R2)
5031  020604  104400                        HLT
5032  020606  106512                        MFPD    (R2)            ;NOTE: MAY BE RESERVED
5033  020610  104400                        HLT
5034  020612  170412                        CLRF    (R2)
5035  020614  104400                        HLT
5036  020616  042202                        BIC     (R2)+,R2
5037  020620  104400                        HLT
5038  020622  164202                        SUB     -(R2),R2
5039  020624  104400                        HLT
5040  020626  155202                        BISB    @-(R2),R2
5041  020630  104400                        HLT
5042  020632  105532                        ADCB    @(R2)+
5043  020634  104400                        HLT
5044  020636  163302                        SUB     @(R3)+,R2
5045  020640  104400                        HLT
5046  020642  005733                        TST     @(R3)+
5047  020644  104400                        HLT
5048  020646  106533                        MFPD    @(R3)+
5049  020650  104400                        HLT
5050  020652  170453                        CLRD    @-(R3)
5051  020654  104400                        HLT
5052  020656  137702  177775                BITB    @.+1,R2
5053  020662  104400                        HLT
5054  020664  105477  177773                NEGB    @.-1
5055  020670  104400                        HLT
5056  020672  000406                        BR      2$
5057
5058  020674  062716  000002        1$:     ADD     #2,(SP)         ;ADJUST RETURN PC
5059  020700  052766  000017  000002        BIS     #17,2(SP)       ;SET CONDITION CODES ON RETURN
5060  020706  000002                        RTI
5061
5062  020710  012706  000500        2$:     MOV     #STKPTR,SP      ;RESET STACK PTR
5063  020714  012737  005540  000004        MOV     #ERPRT,@#ERRVEC ;RESET TIME OUT VECTOR
5064  020722  012737  005530  000010        MOV     #RESERR,@#RESVEC
5065  020730  104000                        SCOPE
```

```
5066
5067                                    ;CHECK JMP INSTRUCTIONS
5068
5069  020732  010700                    MOV     PC,R0
5070  020734  062700  000012            ADD     #12,R0          ;SET ADDRESS FOR JMP INST
5071  020740  000277                    SCC                     ;SET CC'S
5072  020742  000110                    JMP     (R0)
5073  020744  000402                    BR      .+6
5074  020746  000250                    CLN                     ;JMP INST JUMPS HERE
5075  020750  000775                    BR      .-4
5076
5077  020752  103003                    BCC     JMP1
5078  020754  102002                    BVC     JMP1
5079  020756  001001                    BNE     JMP1
5080  020760  100001                    BPL     .+4
5081  020762  104400            JMP1:   HLT                     ;ERROR! INCORRECT CC'S AFTER JMP
5082
5083  020764  005002                    CLR     R2              ;SET INDICATOR
5084  020766  010703                    MOV     PC,R3
5085  020770  000401                    BR      .+4             ;RESERVE WORD FOR JMP ADDRESS
5086  020772  000000                    .WORD   0               ;CONTAINS ADDRESS FOR JMP INST
5087  020774  005723                    TST     (R3)+
5088  020776  010313                    MOV     R3,(R3)
5089  021000  010300                    MOV     R3,R0
5090  021002  062713  000022            ADD     #22,(R3)        ;(R3) IS JMP ADDRESS
5091  021006  010300                    MOV     R3,R0
5092  021010  000133                    JMP     @(R3)+          ;JUMP TO ADDRESS CONTAINED IN R3
5093  021012  000402                    BR      .+6
5094  021014  005102                    COM     R2              ;COMPLEMENT INDICATOR
5095  021016  000775                    BR      .-4
5096  021020  005202                    INC     R2              ;CHECK INDICATOR
5097  021022  001003                    BNE     JMP3
5098  021024  005720                    TST     (R0)+
5099  021026  020003                    CMP     R0,R3           ;CHECK AUTO-INC R3
5100  021030  001401                    BEQ     .+4
5101  021032  104400            JMP3:   HLT
5102
5103  021034  005002                    CLR     R2              ;SET INDICATOR
5104  021036  010704                    MOV     PC,R4           ;SET UP JMP REGISTER
5105  021040  010400                    MOV     R4,R0           ;SET UP CHECK REGISTER
5106  021042  000402                    BR      1$
5107  021044  005102                    COM     R2              ;COMPLEMENT INDICATOR
5108  021046  000403                    BR      2$
5109  021050  022424            1$:     CMP     (R4)+,(R4)+
5110  021052  005724                    TST     (R4)+           ;R4=JMP ADDRESS
5111  021054  000144                    JMP     -(R4)           ;USE R4 AS ADDRESS
5112  021056  005202            2$:     INC     R2              ;CHECK INDICATOR
5113  021060  001003                    BNE     JMP4
5114  021062  022020                    CMP     (R0)+,(R0)+
5115  021064  020004                    CMP     R0,R4           ;CHECK AUTO-DEC R4
5116  021066  001401                    BEQ     .+4
5117  021070  104400            JMP4:   HLT
5118
5119  021072  010703                    MOV     PC,R3
5120  021074  000401                    BR      .+4             ;RESERVE WORD FOR JMP ADDRESS
5121  021076  000000            1$:     .WORD   0               ;CONTAINS JUMP ADDRESS
```

```
5122  021100  005723                          TST     (R3)+
5123  021102  010313                          MOV     R3,(R3)
5124  021104  062723  000016                  ADD     #16,(R3)+
5125  021110  010300                          MOV     R3,R0           ;LOAD CHECK REGISTER
5126  021112  000402                          BR      3$
5127  021114  005102                  2$:     COM     R2
5128  021116  000401                          BR      4$
5129  021120  000153                  3$:     JMP     @-(R3)          ;JUMP TO 2$ VIA 1$ ABOVE
5130  021122  005202                  4$:     INC     R2              ;CHECK INDICATOR
5131  021124  001003                          BNE     JMP5
5132  021126  005740                          TST     -(R0)
5133  021130  020003                          CMP     R0,R3           ;CHECK AUTO-DEC R3
5134  021132  001401                          BEQ     .+4
5135  021134  104400                  JMP5:   HLT
5136
5137  021136  000402                          BR      2$
5138  021140  005102                  1$:     COM     R2              ;COMPLEMENT INDICATOR
5139  021142  000402                          BR      3$
5140  021144  000167  177770          2$:     JMP     1$
5141  021150  005202                  3$:     INC     R2
5142  021152  001401                          BEQ     .+4
5143  021154  104400                  JMP6:   HLT
5144
5145  021156  012767  021174  000020          MOV     #1$,7$          ;SET UP JMP ADDRESS
5146  021164  063767  001004  000012          ADD     @#FACTOR,7$     ;ADD RELOCATION FACTOR
5147  021172  000402                          BR      2$              ;GO TO JMP  @7$ INST
5148  021174  005102                  1$:     COM     R2              ;COMPLEMENT INDICATOR
5149  021176  000403                          BR      3$              ;GO TO CHECK ROUTINE
5150  021200  000177  000000          2$:     JMP     @7$             ;JMP TO 1$ ABOVE VIA 7$
5151  021204  000000                  7$:     .WORD   0               ;CONTAINS JMP ADDRESS
5152  021206  005202                  3$:     INC     R2              ;CHECK INDICATOR
5153  021210  001401                          BEQ     .+4
5154  021212  104400                  JMP7:   HLT
5155  021214  104000                          SCOPE
5156
5157                                  ;CHECK JSR INSTRUCTIONS
5158  021216  013705  001004          JSRTST: MOV     @#FACTOR,R5     ;GET RELOCATION FACTOR
5159  021222  012702  021254                  MOV     #3$,R2          ;FORM DEST ADRS
5160  021226  060502                          ADD     R5,R2           ;ADD RELOCATION FACTOR
5161  021230  000277                          SCC                     ;PRESET CC'S
5162  021232  000242                          CLV
5163  021234  004512                          JSR     R5,(R2)         ;GO TO 3$ VIA R2
5164  021236  005702                  1$:     TST     R2              ;CHECK INDICATOR
5165  021240  001017                          BNE     JSR1            ;R2 SHOULD=0
5166  021242  023705  001004                  CMP     @#FACTOR,R5     ;CHECK THAT RTS  R5 RESTORED R5
5167  021246  001014                          BNE     JSR1
5168  021250  000414                          BR      JSR1A           ;EXIT TO SCOPE
5169  021252  000205                  2$:     RTS     R5              ;RETURN FROM SUBROUTINE
5170  021254  103011                  3$:     BCC     JSR1            ;CHECK THAT JSR DID NOT
5171  021256  102410                          BVS     JSR1
5172  021260  001007                          BNE     JSR1            ;AFFECT CC'S
5173  021262  100006                          BPL     JSR1
5174  021264  005002                          CLR     R2              ;CLEAR INDICATOR
5175  021266  012704  021236                  MOV     #1$,R4          ;GET UNRELOCATED RETURN ADDRESS
5176  021272  061604                          ADD     (SP),R4         ;ADD RELOCATION FACTOR (OLD R5)
5177  021274  020405                          CMP     R4,R5           ;CHECK THAT OLD R5 WAS PLACED ON THE
```

```
5178  021276  001765                    BEQ     2$              ;STACK,& THAT NEW R5 CONTAINS RETURN PC
5179  021300  104400            JSR1:   HLT                     ;ERROR! ABOVE
5180
5181  021302  013704  001004    JSR1A:  MOV     @#FACTOR,R4     ;GET RELOCATION FACTOR
5182  021306  005000                    CLR     R0              ;SET INDICATOR
5183  021310  012705  021330            MOV     #1$,R5
5184  021314  060405                    ADD     R4,R5           ;SET UP JSR DEFERRED ADRS
5185  021316  010502                    MOV     R5,R2
5186  021320  012715  021345            MOV     #5$,(R5)
5187  021324  060415                    ADD     R4,(R5)         ;(R5)=DEST ADRS
5188  021326  000401                    BR      2$              ;RESERVE WORD FOR ADDRESS
5189  021330  000000            1$:     .WORD   0               ;CONTAINS DEST ADRS FOR JSR
5190  021332  004435            2$:     JSR     R4,@(R5)+       ;JSR TO 5$ VIA 1$ ABOVE
5191  021334  005200            3$:     INC     R0              ;CHECK INDICATOR
5192  021336  001013                    BNE     JSR3
5193  021340  000413                    BR      JSR3A
5194  021342  005100            4$:     COM     R0              ;COMPLEMENT INDICATOR
5195  021344  000204                    RTS     4               ;RETURN FROM SUBROUTINE
5196  021346  012703  021334    5$:     MOV     #3$,R3          ;GET UNRELOCATED RETURN ADDRESS
5197  021352  061603                    ADD     (SP),R3         ;ADD RELOCATION FACTOR (OLD R4)
5198  021354  020403                    CMP     R4,R3
5199  021356  001003                    BNE     JSR3
5200  021360  005722                    TST     (R2)+
5201  021362  020205                    CMP     R2,R5           ;CHECK AUTO-INC R5
5202  021364  001766                    BEQ     4$              ;GO TO RTS
5203  021366  104400            JSR3:   HLT                     ;ERROR ABOVE
5204
5205  021370  013704  001004    JSR3A:  MOV     @#FACTOR,R4
5206  021374  010405                    MOV     R4,R5
5207  021376  010703                    MOV     PC,R3
5208  021400  000401                    BR      2$
5209  021402  000405            1$:     BR      4$
5210  021404  022323            2$:     CMP     (R3)+,(R3)+
5211  021406  000277                    SCC
5212  021410  004443                    JSR     R4,-(R3)        ;GO TO 2$
5213  021412  104400            3$:     HLT
5214  021414  000414                    BR      JSR4A
5215  021416  103012            4$:     BCC     JSR4
5216  021420  102011                    BVC     JSR4
5217  021422  001010                    BNE     JSR4
5218  021424  100007                    BPL     JSR4
5219  021426  012702  021412            MOV     #3$,R2          ;GET UNRELOCATED RETURN ADDRESS
5220  021432  061602                    ADD     (SP),R2         ;ADD RELOCATION FACTOR (OLD R4)
5221  021434  020204                    CMP     R2,R4           ;CHECK THAT CALCULATED RETURN
5222  021436  001002                    BNE     JSR4            ;PC = NEW R4
5223  021440  005724                    TST     (R4)+
5224  021442  000204                    RTS     R4
5225  021444  104400            JSR4:   HLT
5226
5227
5228  021446  000401            JSR4A:  BR      2$
5229  021450  000405            1$:     BR      3$
5230  021452  010700            2$:     MOV     PC,R0
5231  021454  004767  177770            JSR     PC,1$
5232  021460  100407                    BMI     JSR6A
5233  021462  104400                    HLT
```

```
5234  021464  022020                  3$:     CMP     (R0)+,(R0)+
5235  021466  020016                          CMP     R0,(SP)         ;CHECK THAT RETURN ADDRESS IS ON THE
5236  021470  001401                          BEQ     .+4             ;STACK
5237  021472  104400                          HLT
5238  021474  000270                          SEN                     ;SET N
5239  021476  000207                          RTS     PC
5240  021500  104000                  JSR6A:  SCOPE
5241
5242                                  ;CHECK IOT TRAP (AND ROLB/ASLB)
5243                                  ;THIS TEST CHECKS THAT THE PSW IS CORRECT AFTER THE IOT AND THAT THE
5244                                  ;'NEW'PSW (FROM IOTVEC+2) IS CORRECT.
5245  021502  012705  000020          IOTTST: MOV     #IOTVEC,R5              ;SET R5=ADDRESS OF IOTVECTOR
5246  021506  010746                          MOV     PC,-(SP)
5247  021510  062716  000040                  ADD     #1$-.,(SP)
5248  021514  012625                          MOV     (SP)+,(R5)+             ;LOAD IOT TRAP VECTOR
5249  021516  005000                          CLR     R0
5250  021520  052740  000200                  BIS     #PRTY4,-(R0)            ;SET PRIORITY LEVEL 4 IN PSW
5251                                                                          ;PSW=X XXX X00 001 1X1 000
5252  021524  011015                          MOV     (R0),(R5)               ;SET IOTVEC+2=PSW ABOVE
5253  021526  011504                          MOV     (R5),R4                 ;SAVE IN R4
5254  021530  042710  000357                  BIC     #PRTY7+17,(R0)
5255  021534  052710  000144                  BIS     #PRTY3+Z,(R0)           ;PSW=X XXX X00 001 1X1 000
5256  021540  012003                          MOV     (R0)+,R3                ;R3 = PSW ABOVE
5257  021542  010340                          MOV     R3,-(R0)
5258  021544  000004                          IOT
5259  021546  104400                  10$:    HLT                             ;ERROR! IOT FAILED TO TRAP
5260
5261  021550  012002                  1$:     MOV     (R0)+,R2                ;GET PSW AFTER IOT TRAP
5262                                                                          ;NOTE: R0=0
5263  021552  012715  000200                  MOV     #PRTY4,(R5)             ;RESTORE IOTVEC+2
5264  021556  012745  002736                  MOV     #.TYPE,-(R5)            ;AND IOTVEC
5265  021562  010746                          MOV     PC,-(SP)                ;FORM PC OF 10$ ABOVE
5266  021564  062716  177762                  ADD     #10$-.,(SP)
5267  021570  022626                          CMP     (SP)+,(SP)+             ;CHECK RETURN PC ON STACK
5268  021572  001036                          BNE     99$
5269  021574  022603                          CMP     (SP)+,R3                ;CHECK SAVED PSW
5270  021576  001034                          BNE     99$
5271  021600  032703  140000                  BIT     #UM,R3                  ;BRANCH TO 3$ IF IN USER MODE
5272  021604  100413                          BMI     3$
5273  021606  001003                          BNE     2$                      ;BRANCH TO 2$ IF IN SUPER MODE
5274  021610  020204                          CMP     R2,R4                   ;CHECK PSW AFTER IOT
5275  021612  001026                          BNE     99$
5276  021614  000413                          BR      4$
5277
5278  021616  042704  030000          2$:     BIC     #PUM,R4                 ;CLEAR PREV MODE BITS
5279  021622  052704  010000                  BIS     #PSM,R4                 ;SET PREV SUPER MODE
5280  021626  020204                          CMP     R2,R4                   ;CHECK PSW AFTER IOT
5281  021630  001017                          BNE     99$
5282  021632  000404                          BR      4$
5283
5284  021634  052704  030000          3$:     BIS     #PUM,R4                 ;SET PREV USER MODE
5285  021640  020204                          CMP     R2,R4                   ;CHECK PSW AFTER IOT
5286  021642  001012                          BNE     99$
5287
5288  021644  005002                  4$:     CLR     R2
5289  021646  000261                          SEC
```

```
5290  021650  106100                  RULB    R0                      ;ROTATE R0
5291  021652  102376                  BVC     .-2                     ;UNTIL V SETS (R0=200)
5292
5293  021654  106300                  ASLB    R0                      ;SHIFT SHOULD SET CARRY
5294  021656  103004                  BCC     99$
5295  021660  102003                  BVC     99$
5296  021662  001002                  BNE     99$
5297  021664  005700                  TST     R0
5298  021666  001401                  BEQ     .+4
5299  021670  104400          99$:    HLT                             ;ERROR! ROL/ASL FAILED TO SET
5300                                                                  ;CC'S PROPERLY (IF R2=0) OR IN-
5301                                                                  ;CORRECT PSW AFTER IOT (IF R2 NOT 0)
5302  021672  042704  000340          BIC     #PRTY7,R4
5303  021676  010437  177776          MOV     R4,@#PSW                ;RESTORE PSW
5304  021702  012706  000500          MOV     #STKPTR,SP              ;RESTORE STACK PTR
5305  021706  104000                  SCOPE
5306
5307                                  ;CHECK EMT TRAP SEQUENCE
5308  021710  005000                  CLR     R0
5309  021712  010746                  MOV     PC,-(SP)
5310  021714  062716  000030          ADD     #EMT1-.,(SP)
5311  021720  012637  000030          MOV     (SP)+,@#EMTVEC
5312  021724  000262                  SEV                             ;SET V
5313  021726  013737  177776  000032  MOV     @#PSW,@#EMTVEC+2        ;RETAIN CURRENT PSW ON TRAP
5314  021734  000265                  +SEZ!SEC
5315  021736  104000                  EMT                     ;TRAP TO EMT1
5316  021740  001433                  BEQ     EMT1C           ;GO TO EMT1C
5317  021742  104400                  HLT                     ;ERROR! INCORRECT CC'S WERE SET ON RETURN
5318  021744  102027          EMT1:   BVC     EMT1B           ;'V' SHOULD'VE SET ON EMT TRAP
5319  021746  105100                  COMB    R0              ;R0=000377,CC'S=1001
5320  021750  105500                  ADCB    R0              ;R0=000000,CC'S=0101
5321  021752  106000                  RORB    R0              ;R0=000200,CC'S=1010
5322  021754  102023                  BVC     EMT1B
5323  021756  100022                  BPL     EMT1B
5324  021760  000257                  CCC
5325  021762  105400                  NEGB    R0              ;R0=000200,CC'S=1010
5326  021764  102017                  BVC     EMT1B
5327  021766  100016                  BPL     EMT1B
5328  021770  000242                  CLV                     ;CLEAR 'V'
5329  021772  000261                  SEC                     ;AND SET 'C'
5330  021774  105300                  DECB    R0              ;R0=000177,CC'S=0011
5331  021776  102012                  BVC     EMT1B
5332  022000  100411                  BMI     EMT1B
5333  022002  000242                  CLV                     ;CLEAR 'V'
5334  022004  105200                  INCB    R0              ;R0=000200,CC'S=1011
5335  022006  103006                  BCC     EMT1B
5336  022010  102005                  BVC     EMT1B
5337  022012  100004                  BPL     EMT1B
5338  022014  000242                  CLV                     ;CLEAR 'V'
5339  022016  106200                  ASRB    R0              ;SHIFT R0 UNTIL 'V' CLEARS
5340  022020  102776                  BVS     .-2
5341  022022  000401                  BR      .+4
5342  022024  104400          EMT1B:  HLT                     ;ERROR!
5343  022026  000002                  RTI                     ;EXIT WITH R0=000377
5344  022030  105500          EMT1C:  ADCB    R0              ;R0=000000
5345  022032  103003                  BCC     EMT1D
```

```
5346  022034  001002                          BNE     EMT1D
5347  022036  005700                          TST     R0
5348  022040  001401                          BEQ     .+4
5349  022042  104400                  EMT1D:  HLT
5350  022044  012737  001014  000030          MOV     #SCOPEA,@#EMTVEC        ;RESTORE EMT TO SCOPE
5351  022052  005037  000032                  CLR     @#EMTVEC+2
5352  022056  104000                          SCOPE
5353
5354                                  ;CHECK TRAP INSTRUCTION TRAP SEQUENCE
5355          104000                          HLT=EMT                         ;REDEFINE HLT
5356  022060  013737  000034  000030          MOV     @#TRAPVEC,@#EMTVEC      ;SET EMT (HLT) TRAP VECTOR
5357  022066  010746                          MOV     PC,-(SP)
5358  022070  062716  000042                  ADD     #TRAP1-.,(SP)
5359  022074  012637  000034                  MOV     (SP)+,@#TRAPVEC
5360  022100  000270                          SEN                             ;SET N
5361  022102  013737  177776  000036          MOV     @#PSW,@#TRAPVEC+2       ;RETAIN CURRENT PSW ON TRAP
5362  022110  000261                          SEC                             ;SET CARRY
5363  022112  010700                          MOV     PC,R0
5364  022114  000264                          SEZ                     ;SET Z BIT
5365  022116  104400                          TRAP                    ;TRAP TO TRAP1
5366  022120  103401                          BCS     .+4
5367  022122  104000                          HLT
5368  022124  001401                          BEQ     .+4
5369  022126  104000                          HLT
5370  022130  000412                          BR      TRAP1C
5371  022132  100401                  TRAP1:  BMI     .+4                     ;N BIT GOT SET ON TRAP
5372  022134  104000                          HLT
5373  022136  062700  000004                  ADD     #4,R0
5374  022142  020016                          CMP     R0,(SP)                 ;CHECK LOW BYTE OF RETURN PC ON
5375  022144  001401                          BEQ     .+4                     ;STACK
5376  022146  104000                          HLT
5377  022150  124646                          CMPB    -(SP),-(SP)
5378  022152  032626                          BIT     (SP)+,(SP)+
5379  022154  000002                          RTI                             ;RETURN TO INST FOLLOWING TRAP (1$)
5380
5381  022156  012702  000036          TRAP1C: MOV     #TRAPVEC+2,R2                   ;RESTORE VECTORS
5382  022162  012712  000340                  MOV     #PRTY7,(R2)
5383  022166  012742  003416                  MOV     #.HLT,-(R2)
5384  022172  005042                          CLR     -(R2)
5385  022174  012742  001014                  MOV     #SCOPEA,-(R2)
5386  022200  104000                          SCOPE
5387          104400                          HLT=TRAP                        ;RESTORE HLT TO A TRAP INST
5388
5389  022202  010702                          MOV     PC,R2
5390  022204  062702  000012                  ADD     #12,R2
5391  022210  012707  001152                  MOV     #RELOC,PC               ;GO RELOCATE PROGRAM CODE
5392  022214  000000                  REL33:  .WORD   0
5393                                  ;3333333333333 LAST ADDRESS OF CODE TO BE RELOCATED 33333333333
5394
5395  022216  010701                          MOV     PC,R1                   ;SET SCOPE POINTER
5396  022220  122737  000004  000766          CMPB    #4,@#OPT.CP                     ;BRANCH IF 11/40 OR 11/45
5397  022226  101405                          BLOS    REL4
5398  022230  012737  000002  031620          MOV     #RTI,@#RTI1                     ;SET 'T' TRAP RETURN TO RTI
5399  022236  000137  030466                  JMP     @#TTYCHK                        ;JUMP IF 11/05 OR 11/20
5400
5401                                          .SBTTL  START OF SECTION 4
```

```
5402                                    ;444444444444 FIRST ADDRESS TO BE RELOCATED 444444444
5403  022242  010700                    REL4:   MOV     PC,R0           ;GET PC
5404  022244  005740                            TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL4
5405  022246  010037  001010                    MOV     R0,@#FRSTAD     ;SAVE
5406  022252  012737  000004  005442            MOV     #4,@#SECT       ;SET SECTION #
5407  022260  004737  005432                    JSR     PC,@#LDDISP     ;LOAD DISPLAY REG
5408  022264  013767  005436  001370            MOV     @#DISPLY,REL44
5409  022272  010700                            MOV     PC,R0           ;GET CURRENT PC
5410  022274  162700  022274                    SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
5411  022300  010037  001004                    MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
5412  022304  010701                            MOV     PC,R1           ;SET NEW SCOPE PTR
5413
5414                                    ;CHECK STACK OVERFLOW
5415  022306  013767  177776  000332    OVFLW:  MOV     @#PSW,7$        ;SAVE STATUS IN 7$ BELOW
5416  022314  005037  177776                    CLR     @#PSW           ;SET KERNEL MODE
5417  022320  004737  002676                    JSR     PC,@#CLRTBIT    ;GO CLEAR '1' BIT IF SET
5418  022324  052737  000340  177776            BIS     #PRTY7,@#PSW    ;SET PRIORITY LEVEL 7 TO BLOCK CLOCK
5419  022332  010746                            MOV     PC,-(SP)        ;PUSH CURRENT PC ONTO STACK
5420  022334  062716  000146                    ADD     #2$-.,(SP)      ;FORM ADDRESS OF 2$ BELOW
5421  022340  011637  000004                    MOV     (SP),@#ERRVEC   ;SET ERROR VECTOR
5422  022344  012737  000340  000006          MOV     #340,@#ERRVEC+2 ;SET PRIORITY LEVEL 7 ON TRAP
5423  022352  013727  000016                    MOV     @#BPTVEC+2,(PC)+        ;SAVE CONTENTS OF BPT VECTOR +2
5424  022356  000000                    42$:    .WORD   0
5425  022360  062716  000100                    ADD     #41$-2$,(SP)    ;FORM ADDRESS OF 41$ BELOW
5426  022364  012637  000014                    MOV     (SP)+,@#BPTVEC  ;SET BPT TRAP VECTOR TO 41$
5427  022370  012737  000340  000016          MOV     #340,@#BPTVEC+2
5428
5429  022376  012703  000376                    MOV     #376,R3
5430  022402  010313                            MOV     R3,(R3)         ;LOAD 376 INTO ADDRESS 376
5431  022404  010306                            MOV     R3,SP           ;SET STACK PTR AT BOUNDARY
5432  022406  032767  140000  000232            BIT     #UM,7$          ;CHECK IF ENTERED TEST IN KERNEL
5433  022414  001015                            BNE     1$              ;MODE. BRANCH IF NOT IN KERNEL
5434
5435                                    ;THE BELOW INSTRUCTIONS SHOULD NOT CAUSE AN OVERFLOW TRAP
5436  022416  005716                            TST     (SP)            ;BECAUSE TST IS A NON MODIFYING INST
5437  022420  021666  177776                    CMP     (SP),-2(SP)     ;SO IS COMPARE
5438  022424  012656                            MOV     (SP)+,@-(SP)    ;BECAUSE OF ADDRESS MODE 5
5439  022426  057636  000000                    BIS     @(SP),@(SP)+    ;BECAUSE OF ADDRESS MODE 3
5440  022432  057676  000000                    BIS     -(SP),@(SP)     ;BECAUSE OF ADDRESS MODE 7
5441  022436  005006                            CLR     SP
5442  022440  013766  020000  020000            MOV     @#20000,20000(SP)
5443  022446  000423                            BR      3$              ;BRANCH OVER NON KERNEL MODE TESTS
5444
5445                                    ;NOTE:  NO OVEFLOW TRAP WILL OCCUR IF NOT IN KERNEL MODE!!!
5446  022450  156737  000173  177777    1$:     BISB    7$+1,@#PSW+1    ;RESTORE MODE BITS IN PSW
5447  022456  012706  000376                    MOV     #376,SP         ;SET STACK PTR
5448  022462  016646  177776                    MOV     -2(SP),-(SP)    ;SHOULD NOT TRAP
5449  022466  051616                            BIS     (SP),(SP)
5450  022470  061666  177776                    ADD     (SP),-2(SP)
5451  022474  105037  177777                    CLRB    @#PSW+1         ;SET KERNEL MODE
5452  022500  000451                            BR      6$              ;EXIT TEST
5453
5454                                    ;ERROR SERVICE ROUTINE
5455  022502  012600                    2$:     MOV     (SP)+,R0        ;SAVE PC OF INSTRUCTION THAT TRAPPED
5456  022504  012602                            MOV     (SP)+,R2        ;SAVE PSW
5457  022506  012706  000500                    MOV     #STKPTR,SP      ;SET STACK PTR
```

```
5458  022512  104400                          HLT                     ;ERROR! AN INSTRUCTION THAT WAS NOT
5459                                                                  ;SUPPOESED TO TRAP TRAPPED
5460                                                                  ;R0 CONTAINS PC, R2 CONTAINS PSW
5461  022514  000443                          BR      6$              ;EXIT TEST
5462                                  ;THE BELOW INSTRUCTIONS WILL CAUSE A STACK OVERFLOW
5463                                  ;STACK PTR IS AT 376
5464  022516  062737  000066  000004  3$:     ADD     #4$-2$,@#ERRVEC ;SET ERROR VECTOR TO 4$
5465  022524  010306                          MOV     R3,SP           ;SET STACK PTR AT 376
5466  022526  112702  000001                  MOVB    #1,R2
5467  022532  005000                          CLR     R0
5468  022534  005016                          CLR     (SP)            ;SETS BIT 0 IN R0
5469  022536  006302                          ASL     R2              ;SHIFT INDICATOR BIT
5470  022540  105226                          INCB    (SP)+           ;SETS BIT 1 IN R0
5471  022542  006302                          ASL     R2
5472  022544  060746                          ADD     PC,-(SP)        ;SETS BIT 2 IN R0
5473  022546  006302                          ASL     R2
5474  022550  000003                          BPT                     ;SETS BIT 3 IN R0
5475  022552  006302                          ASL     R2
5476  022554  004767  000014                  JSR     PC,40$          ;SETS BIT 4 IN R0
5477  022560  006302                          ASL     R2
5478  022562  050666  177776                  BIS     SP,-2(SP)       ;SETS BIT 5 IN R0
5479  022566  000410                          BR      5$
5480
5481                                  ;PROGRAM WILL TRAP HERE ON OVERFLOW TRAP
5482  022570  050200                  4$:     BIS     R2,R0           ;SET APPROPRIATE BIT IN R0
5483  022572  000002                          RTI                     ;RETURN FROM TRAP
5484
5485  022574  052700  001000          40$:    BIS     #1000,R0                ;SET IND THAT JSR WAS EXECUTED
5486  022600  000207                          RTS     PC
5487
5488  022602  052700  000400          41$:    BIS     #400,R0                 ;SET IND THAT BPT WAS EXECUTED
5489  022606  000002                          RTI
5490
5491                                  ;CHECK THAT ABOVE INSRUCTIONS DID TRAP
5492  022610  012706  000500          5$:     MOV     #STKPTR,SP      ;SET STACK PTR
5493  022614  022700  001477                  CMP     #1477,R0        ;EACH INSTRUCTION SET A BIT IN R0
5494  022620  001401                          BEQ     .+4             ;R0= 1477
5495  022622  104400                          HLT
5496
5497                                  ;EXIT ROUTINE
5498  022624  012706  000600          6$:     MOV     #KPTR,SP        ;SET KERNEL STACK PTR
5499  022630  012737  000016  000014          MOV     #BPTVEC+2,@#BPTVEC
5500  022636  016737  177514  000016          MOV     42$,@#BPTVEC+2
5501  022644  012746                          MOV     (PC)+,-(SP)     ;PUSH OLD PSW ONTO STACK
5502  022646  000000                  7$:     .WORD   0               ;CONTAINS SAVED PSW
5503  022650  010746                          MOV     PC,-(SP)        ;PUSH CURRENT PC ONTO STACK
5504  022652  062716  000006                  ADD     #6,(SP)         ;ADD OFFSET
5505  022656  000002                          RTI
5506  022660  012706  000500                  MOV     #STKPTR,SP      ;SET STACK PTR
5507  022664  012737  005540  000004          MOV     #ERPRT,@#ERRVEC ;RESET TIME OUT VECTOR
5508  022672  012737  000002  000006          MOV     #RTI,@#ERRVEC+2
5509  022700  104000                          SCOPE
5510
5511                                  ;CHECK THAT ALL RESERVED INSTRUCTIONS TRAP (TO LOCATION 10)
5512  022702  012702  023006          RESTRP: MOV     #5$,R2          ;GET ADDRESS OF RESERVED INSTRUCTION TABLE
5513  022706  063702  001004                  ADD     @#FACTOR,R2
```

```
5514  022712  132737  000040  000767          BITB    #40,@#OPT.CP+1  ;CHECK IF 11/45 FLOATING POINT IS AVAIL.
5515  022720  001402                          BEQ     .+6             ;BRANCH IF NOT AVAILABLE
5516  022722  005067  000110                  CLR     50$             ;SET TABLE TERMINATOR AT GROUP 7
5517  022726  012737  022764  000010          MOV     #4$,@#RESVEC    ;SET RESERVED INSTRUCTION TRAP
5518  022734  063737  001004  000010          ADD     @#FACTOR,@#RESVEC
5519  022742  012203                  1$:     MOV     (R2)+,R3        ;GET FIRST RESERVED INSTRUCTION
5520  022744  001437                          BEQ     7$              ;0 TERMINATES THE TABLE
5521  022746  012204                          MOV     (R2)+,R4        ;GET LAST RESERVED INSTRUCTION IN GROUP
5522  022750  010317                  2$:     MOV     R3,(PC)         ;EXECUTE RESERVED INSTRUCTION
5523  022752  000000                  3$:     .WORD   0               ;CONTAINS RESERVED INSTRUCTION
5524  022754  104400                          HLT                     ;ERROR! INSTRUCTION IN R3
5525  022756  104400                          HLT                     ;(2$) ABOVE FAILED TO CAUSE A
5526  022760  104400                          HLT                     ;RESERVED INSTRUCTION TRAP
5527  022762  000405                          BR      41$
5528  022764  012716  022776          4$:     MOV     #41$,(SP)       ;ADJUST RETURN PC
5529  022770  063716  001004                  ADD     @#FACTOR,(SP)   ;TO RETURN TO 41$
5530  022774  000002                          RTI                     ;RETURN TO 41$
5531  022776  020304                  41$:    CMP     R3,R4           ;HAS GROUP OF RESERVED INSTRUCTIONS
5532  023000  001760                          BEQ     1$              ;BEEN EXECUTED
5533  023002  005203                          INC     R3              ;INCREMENT THIS RESERVED INSTRUCTION
5534  023004  000761                          BR      2$              ;TO NEXT ONE AND EXECUTE
5535                                  ;TABLE OF 11/40,11/45 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)
5536  023006  000007                  5$:     7                       ;GROUP 1
5537  023010  000077                          77                      ;  "
5538  023012  000210                          210                     ;GROUP 2
5539  023014  000227                          227                     ;  "
5540  023016  007000                          7000                    ;GROUP 3
5541  023020  007777                          7777                    ;  "
5542  023022  075040                          75040                   ;GROUP 4
5543  023024  076777                          76777                   ;  "
5544  023026  106400                          106400                  ;GROUP 5
5545  023030  106477                          106477                  ;  "
5546  023032  106700                          106700                  ;GROUP 6
5547  023034  107777                          107777                  ;  "
5548  023036  170000                  50$:    170000                  ;GROUP 7        FLOATING POINT
5549  023040  177777                          177777                  ;  "            INSTRUCTIONS
5550  023042  000000                          0                       ;0 TERMINATES THE TABLE
5551
5552  023044  012737  005530  000010  7$:     MOV     #RESERR,@#RESVEC        ;RESTORE RESERVED TRAP
5553  023052  104000                          SCOPE
5554
5555                                  ;CHECK THAT ALL BITS IN THE PROCESSER STATUS WORD (PSW) CAN BE SET AND
5556                                  ;CLEARED.
5557  023054  105737  000770          PSWCHK: TSTB    @#MMON          ;IF MEM MGMT IS ON SKIP THIS TEST
5558  023060  001072                          BNE     4$
5559  023062  013767  177776  000144          MOV     @#PSW,3$        ;SAVE STATUS
5560  023070  005037  177776                  CLR     @#PSW           ;CLEAR MODE BITS IN PSW
5561  023074  004737  002676                  JSR     PC,@#CLRTBIT    ;GO CLEAR 'T' BIT IF SET
5562  023100  013746  000016                  MOV     @#TBITVEC+2,-(SP)
5563  023104  012704  177776                  MOV     #PSW,R4         ;LOAD ADDRESS OF PSW INTO R4
5564  023110  000250                          CLN
5565  023112  005714                          TST     (R4)            ;CHECK THAT PSW WAS CLEARED
5566  023114  001401                          BEQ     .+4
5567  023116  104400                          HLT                     ;ERROR! PSW FAILED TO CLEAR
5568  023120  113700  000766                  MOVB    @#OPT.CP,R0     ;GET CP TYPE
5569  023124  016000  032664                  MOV     PSWBIT(0),R0    ;GET BIT MASK FOR TEST R0=THOSE BITS IN
```

```
5570                                                    ;THE PSW WHICH CAN BE SET/CLEARED.
5571  023130  005737  000766                  TST     @#OPT.CP        ;CHECK IF MEM MGMT IS AVAILABLE
5572  023134  100002                          BPL     10$             ;BRANCH IF NOT AVAILABLE
5573  023136  052700  170000                  BIS     #170000,R0      ;SET BITS 15-12 IF MEM MGMT
5574  023142  012702  000001          10$:    MOV     #1,R2           ;R2 = TEST BIT
5575  023146  030200                  1$:     BIT     R2,R0           ;CHECK IF BIT CAN BE SET/CLEARED
5576  023150  001423                          BEQ     2$
5577  023152  005037  000016                  CLR     @#TBITVEC+2
5578  023156  030227  000020                  BIT     R2,#20          ;CHECK IF TEST WILL SET 'T' BIT
5579  023162  001403                          BEQ     20$
5580  023164  012737  000002  000016          MOV     #RTI,@#TBITVEC+2;SET RTI INTO RETURN
5581  023172  005014                  20$:    CLR     (R4)            ;CLEAR PSW
5582  023174  050214                          BIS     R2,(R4)         ;SET R2 INTO PSW
5583  023176  011403                          MOV     (R4),R3         ;GET BIT
5584  023200  020203                          CMP     R2,R3           ;CHECK THAT BIT WAS SET IN PSW
5585  023202  001401                          BEQ     .+4
5586  023204  104400                          HLT                     ;ERROR! BIT IN R2 FAILED TO SET IN PSW
5587  023206  000244                          CLZ                     ;CLEAR Z BIT
5588  023210  040214                          BIC     R2,(R4)         ;CLEAR BIT IN PSW
5589  023212  011403                          MOV     (R4),R3         ;GET PSW RESULT
5590  023214  001401                          BEQ     2$              ;BRANCH IF BIC ABOVE CLEARED BIT IN PSW
5591  023216  104400                          HLT                     ;ERROR! BIT IN R2 FAILED TO CLEAR IN PSW
5592  023220  006302                  2$:     ASL     R2              ;SHIFT TEST BIT
5593  023222  103351                          BCC     1$              ;BRANCH IF ALL BITS NOT TESTED
5594  023224  005014                          CLR     (R4)            ;CLEAR STATUS
5595  023226  012637  000016                  MOV     (SP)+,@#TBITVEC+2       ;RESTORE T BIT RETURN
5596  023232  012746                          MOV     (PC)+,-(SP)     ;PUSH ORIGINAL STATUS ON STACK
5597  023234  000000                  3$:     .WORD   0               ;CONTAINS ORIGINAL PSW
5598  023236  010746                          MOV     PC,-(SP)        ;SET RETURN PC
5599  023240  062716  000006                  ADD     #6,(SP)
5600  023244  000002                          RTI                     ;RETURN
5601  023246  104000                  4$:     SCOPE
5602
5603  023250  013704  177776                  MOV     @#PSW,R4        ;SAVE PSW IN R4
5604  023254  112737  000300  177776          MOVB    #300,@#PSW      ;SET PRIORITY LEVEL 6
5605  023262  004737  002676                  JSR     PC,@#CLRTBIT    ;GO CLEAR 'T' BIT IF SET
5606
5607                                          ;CHECK THAT ALL BITS IN THE CURRENT STACK PTR CAN BE SET/CLEARED
5608  023266  010603                  CHKSP:  MOV     SP,R3           ;SAVE STACK PTR
5609  023270  000257                          CCC
5610  023272  112706  000377                  MOVB    #377,SP         ;SET STACK PTR = -1
5611  023276  006006                  1$:     ROR     SP              ;ROTATE 0 BIT THROUGH ALL BIT
5612  023300  103776                          BCS     1$              ;BIT POSITIONS
5613  023302  005206                          INC     SP              ;SHOULD INCREMENT SP TO 0
5614  023304  001403                          BEQ     2$
5615  023306  010602                          MOV     SP,R2           ;SAVE ERROR STACK PTR
5616  023310  010306                          MOV     R3,SP           ;SET STACK PTR FOR TRAP
5617  023312  104400                          HLT                     ;ERROR!
5618
5619  023314  010306                  2$:     MOV     R3,SP           ;RESTORE ORIGINAL STACK PTR
5620
5621                                          ;CHECK BYTE OPERATIONS USING THE STACK
5622  023316  010600                  SPCHK:  MOV     SP,R0           ;SAVE STACK PTR
5623  023320  010003                          MOV     R0,R3
5624
5625  023322  005043                          CLR     -(R3)
```

```
5626  023324  112746  177777          MOVB    #-1,(SP)        ;(SP) = 377
5627  023330  022713  000377          CMP     #377,(R3)       ;CHECK THAT ONLY EVEN BYTE WAS AFFECTED
5628  023334  001002                  BNE     1$
5629  023336  020306                  CMP     R3,SP           ;CHECK AUTO-DEC
5630  023340  001401                  BEQ     .+4
5631  023342  104400          1$:     HLT
5632
5633  023344  105226                  INCB    (SP)+
5634  023346  005723                  TST     (R3)+           ;CHECK RESULT
5635  023350  001002                  BNE     2$
5636  023352  020006                  CMP     R0,SP           ;CHECK AUTO-INC
5637  023354  001401                  BEQ     .+4
5638  023356  104400          2$:     HLT
5639
5640  023360  005143                  COM     -(R3)           ;(R3)=177777
5641  023362  144613                  BICB    -(SP),(R3)
5642  023364  022713  177400          CMP     #177400,(R3)    ;CHECK RESULT
5643  023370  001002                  BNE     3$
5644  023372  020603                  CMP     SP,R3
5645  023374  001401                  BEQ     .+4
5646  023376  104400          3$:     HLT
5647
5648  023400  132627  000377          BITB    (SP)+,#377
5649  023404  001002                  BNE     4$
5650  023406  020600                  CMP     SP,R0
5651  023410  001401                  BEQ     .+4
5652  023412  104400          4$:     HLT
5653
5654  023414  012746  000001          MOV     #1,-(SP)
5655  023420  062706  000002          ADD     #2,SP
5656  023424  012702  177401          MOV     #177401,R2
5657  023430  120246                  CMPB    R2,-(SP)
5658  023432  001004                  BNE     5$
5659  023434  122602                  CMPB    (SP)+,R2
5660  023436  001002                  BNE     5$
5661  023440  020006                  CMP     R0,SP
5662  023442  001401                  BEQ     .+4
5663  023444  104400          5$:     HLT
5664  023446  105037  177776          CLRB    @#PSW
5665  023452  010446                  MOV     R4,-(SP)        ;RESTORE ORIGINAL PSW TO STACK
5666  023454  010746                  MOV     PC,-(SP)
5667  023456  062716  000006          ADD     #6,(SP)
5668  023462  000002                  RTI
5669  023464  104000                  SCOPE
5670
5671                          ;CHECK THAT 'C' BIT SETS/CLEARS PROPERLY
5672  023466  012727  177776  CBIT:   MOV     #177776,(PC)+   ;LOAD CONSTANT
5673  023472  000000          1$:     .WORD   0
5674  023474  010700                  MOV     PC,R0           ;GET CURRENT PC
5675  023476  162700  000004          SUB     #4,R0           ;POINT R0 TO 1$ ABOVE
5676  023502  005520          2$:     ADC     (R0)+           ;ADD 'C' BIT TO 1$ ABOVE
5677  023504  006340                  ASL     -(R0)           ;SHIFT 1$
5678  023506  102375                  BVC     2$              ;UNTIL 'V' BIT SETS
5679  023510  022767  077776  177754  CMP     #077776,1$      ;CHECK RESULT
5680  023516  001401                  BEQ     .+4
5681  023520  104400                  HLT                     ;ERROR! INCORRECT RESULT IN 1$ ABOVE
```

```
                                                         ;R0=ADDRESS OF DATA

                         ;CHECK THAT CONDITION CODES ARE SET PROPERLY WHEN A NUMBER (CURRENT PC)
                         ;AND THAT NUMBER +1 ARE COMPARED, AND VICE VERSA.
023522  010700           CMPN:   MOV     PC,R0           ;GET CURRENT PC
023524  010002                   MOV     R0,R2           ;SAVE IN R2
023526  005202                   INC     R2              ;MAKE R2 = R0+1
023530  000277                   SCC
023532  000251                   +CLC!CLN                ;CLEAR C & N BITS
023534  020002                   CMP     R0,R2           ;COMPARE # WITH #+1
023536  103003                   BCC     1$              ;CARRY BIT SHOULD SET
023540  102402                   BVS     1$              ;V BIT SHOULD CLEAR
023542  001401                   BEQ     1$              ;Z BIT SHOULD CLEAR
023544  100401                   BMI     .+4             ;N BIT SHOULD SET
023546  104400           1$:     HLT                     ;ERROR! COMPARE # WITH #+1 FAILED TO
                                                         ;SET CONDITION CODES IN PSW CORRECTLY

023550  000277                   SCC                     ;SET CONDITION CODES IN PSW
023552  120200                   CMPE    R2,R0           ;COMPARE #+1 WITH #
023554  103403                   BCS     2$              ;C BIT SHOULD CLEAR
023556  102402                   BVS     2$              ;V BIT SHOULD CLEAR
023560  001401                   BEQ     2$              ;Z BIT SHOULD CLEAR
023562  100001                   BPL     .+4             ;N BIT SHOLD CLEAR
023564  104400           2$:     HLT                     ;ERROR! COMPARE #+1 WITH # FAILED TO SET
                                                         ;CONDITION CODES IN PSW CORRECTLY

                         ;24 NOP (240) INSTRUCTIONS FOLLOW. THESE NOPS MAY
                         ;BE CHANGED TO TEST CODE IF THE NEED ARISES. THE TEST CODE SHOULD
                         ;BE POSITION INDEPENDENT AND SHOULD RUN WHEN RELOCATED BY THE PROGRAM.
023566  000240                   NOP
023570  000240                   NOP
023572  000240                   NOP
023574  000240                   NOP
023576  000240                   NOP
023600  000240                   NOP
023602  000240                   NOP
023604  000240                   NOP
023606  000240                   NOP
023610  000240                   NOP
023612  000240                   NOP
023614  000240                   NOP
023616  000240                   NOP
023620  000240                   NOP
023622  000240                   NOP
023624  000240                   NOP
023626  000240                   NOP
023630  000240                   NOP
023632  000240                   NOP
023634  000240                   NOP
023636  000240                   NOP
023640  000240                   NOP
023642  000240                   NOP
023644  000240                   NOP
023646  104000                   SCOPE

023650  010702                   MOV     PC,R2
```

```
5738  023652  062702  000012                  ADD     #12,R2
5739  023656  012707  001152                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
5740  023662  000000                  REL44:  .WORD   0
5741                                  ;444444444444444 LAST ADDRESS OF CODE TO BE RELOCATED 444444444444
5742
5743
5744
5745                                          .SBTTL  START OF SECTION 5
5746                                  ;555555555555555 FIRST ADDRESS TO BE RELOCATED 555555555
5747  023664  010700                  REL5:   MOV     PC,R0           ;GET PC
5748  023666  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL5
5749  023670  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
5750  023674  012737  000005  005442          MOV     #5,@#SECT       ;SET SECTION #
5751  023702  004737  005432                  JSR     PC,@#LDDISP     ;LOAD DISPLAY REG
5752  023706  013767  005436  001462          MOV     @#DISPLY,REL55
5753  023714  010700                          MOV     PC,R0           ;GET CURRENT PC
5754  023716  162700  023716                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
5755  023722  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
5756  023726  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR
5757
5758                                  ;CHECK EXTENDED INSTRUCTION SET (SXT, XOR, SOB, MARK, RTI/RTT)
5759  023730  005000                  EXTINST:CLR     R0
5760  023732  000277                          SCC                     ;PRESET CC'S
5761  023734  006700                          SXT     R0              ;EXTEND SIGN (1) INTO R0
5762  023736  103005                          BCC     SXT0            ;CHECK RESULT CC'S
5763  023740  102404                          BVS     SXT0
5764  023742  001403                          BEQ     SXT0
5765  023744  100002                          BPL     SXT0
5766  023746  005200                          INC     R0              ;CHECK RESULT
5767  023750  001401                          BEQ     .+4
5768  023752  104400                  SXT0:   HLT
5769
5770  023754  010700                          MOV     PC,R0
5771  023756  010002                          MOV     R0,R2
5772  023760  012703  177777                  MOV     #-1,R3
5773  023764  005102                          COM     R2
5774  023766  000243                          +CLV!CLC                ;CLEAR C AND V BITS
5775  023770  074003                          XOR     R0,R3           ;R3 SHOULD CONTAIN COMPLEMENT OF R0
5776  023772  103404                          BCS     XORO            ;CHECK THAT C WAS NOT AFFECTED
5777  023774  102403                          BVS     XORO            ;AND THAT V WAS CLEARED
5778  023776  001402                          BEQ     XORO
5779  024000  020203                          CMP     R2,R3           ;CHECK RESULT
5780  024002  001401                          BEQ     .+4
5781  024004  104400                  XORO:   HLT                     ;ERROR!   XOR FAILED
5782
5783  024006  010700                          MOV     PC,R0
5784  024010  022020                          CMP     (R0)+,(R0)+     ;SET ADDRESS REGISTER
5785  024012  000401                          BR      .1$             ;RESERVE WORD FOR TEST DATA
5786  024014  000000                          .WORD   0               ;CONTAINS TEST DATA
5787  024016  005700                  1$:     TST     R0              ;EXTEND SIGN OF ADDRESS INTO
5788  024020  006710                          SXT     (R0)            ;ADDRESS (R0)=-1 IF MSB R0=1
5789  024022  005002                          CLR     R2              ;OTHERWISE, (R0)=0
5790  024024  005700                          TST     R0              ;CHECK SIGN OF ADDRESS
5791  024026  100001                          BPL     .+4
5792  024030  005102                          COM     R2              ;COMPLEMENT CHECK REG IF NEG
5793  024032  021002                          CMP     (R0),R2         ;CHECK RESULT OF SXT
```

```
5794  024034  001401                    BEQ     .+4
5795  024036  104400            SXT1:   HLT                     ;ERROR!   SXT FAILED TO EXTEND SIGN PROPERLY
5796
5797  024040  012710  100000            MOV     #100000,(R0)    ;PRESET DATA
5798  024044  011002                    MOV     (R0),R2
5799  024046  000277                    SCC                     ;PRESET CC'S
5800  024050  074210                    XOR     R2,(R0)         ;XOR 100000 WITH 100000 RESULT = 0
5801  024052  103007                    BCC     XOR1            ;CHECK CC'S AFTER XOR
5802  024054  102406                    BVS     XOR1
5803  024056  001005                    BNE     XOR1
5804  024060  100404                    BMI     XOR1
5805  024062  005710                    TST     (R0)            ;CHECK RESULT (0)
5806  024064  001002                    BNE     XOR1
5807  024066  005402                    NEG     R2              ;CHECK THAT REG WAS NOT AFFECTED
5808  024070  102401                    BVS     .+4
5809  024072  104400            XOR1:   HLT
5810
5811  024074  010702                    MOV     PC,R2
5812  024076  022222                    CMP     (R2)+,(R2)+
5813  024100  000401                    BR      SXT4            ;PRESERVE WORD FOR DATA
5814  024102  000000                    .WORD   0               ;RESERVED FOR DATA
5815  024104  012722  125252    SXT4:   MOV     #125252,(R2)+   ;PRESET DATA
5816  024110  006742                    SXT     -(R2)           ;EXTEND SIGN
5817  024112  074722                    XOR     PC,(R2)+
5818  024114  010700                    MOV     PC,R0           ;GET PC
5819  024116  005740                    TST     -(R0)           ;SUBTRACT 2 FROM PC
5820  024120  005100                    COM     R0              ;R0=RESULT OF XOR PC-1 ABOVE
5821  024122  074042                    XOR     R0,-(R2)        ;CHECK RESULT OF SXT AND XOR ABOVE
5822  024124  001401                    BEQ     .+4
5823  024126  104400            XOR24:  HLT                     ;ERROR!   SXT & XOR ABOVE INCORRECT
5824
5825  024130  012704  000001            MOV     #1,R4           ;SET R4
5826  024134  006767  000060            SXT     XOR6A           ;PRESET DATA=0
5827  024140  074467  000054    2$:     XOR     R4,XOR6A
5828  024144  100423                    BMI     XOR6
5829  024146  006304                    ASL     R4              ;SHIFT R4
5830  024150  102373                    BVC     2$              ;UNTIL V SETS (R4=100000)
5831  024152  100020                    BPL     XOR6            ;BRANCH IF 'N' IS CLEAR
5832  024154  074467  000040            XOR     R4,XOR6A        ;XOR6A=177777
5833  024160  100015                    BPL     XOR6
5834  024162  074767  000032            XOR     PC,XOR6A        ;XOR PC WITH XOR6A (177777)
5835  024166  010767  000030            MOV     PC,XOR6B        ;FORM PC AS USED IN XOR ABOVE
5836  024172  162767  000004  000022    SUB     #4,XOR6B
5837  024200  005167  000016            COM     XOR6B
5838  024204  026767  000012  000006    CMP     XOR6B,XOR6A     ;XOR6A SHOULD = COMPLEMENT OF PC
5839  024212  001401                    BEQ     .+4
5840  024214  104400            XOR6:   HLT                     ;ERROR!  XOR TESTS ABOVE FAILED
5841
5842  024216  000402                    BR      .+6
5843
5844  024220  000000            XOR6A:  .WORD   0               ;CONTAINS DATA USED BY TEST ABOVE
5845  024222  000000            XOR6B:  .WORD   0
5846
5847
5848  024224  012700  077777            MOV     #077777,R0      ;SET SOURCE OPERAND FOR ADD
5849  024230  006767  177764            SXT     XOR6A           ;CLEAR XOR6A
```

```
5850  024234  001004                  BNE     SXT6            ;CHECK CC'S AFTER EXTENDING ZERO'S
5851  024236  100403                  BMI     SXT6
5852  024240  103402                  BCS     SXT6
5853  024242  102401                  BVS     SXT6
5854  024244  000401                  BR      .+4
5855  024246  104400          SXT6:   HLT                     ;ERROR!  SXT FAILED
5856
5857  024250  012702  000001          MOV     #1,R2           ;SET DEST OPERAND FOR ADD
5858  024254  013703  001004          MOV     @#FACTOR,R3     ;LOAD INDEX REGISTER
5859  024260  060002                  ADD     R0,R2           ;RESULT OF ADD=100000
5860  024262  006763  024220          SXT     XOR6A(3)        ;EXTEND SIGN OF ADD ABOVE
5861  024266  001403                  BEQ     SXT6A
5862  024270  005267  177724          INC     XOR6A           ;CHECK RESULT OF SXT
5863  024274  001401                  BEQ     .+4
5864  024276  104400          SXT6A:  HLT                     ;ERROR!  SXT ABOVE FAILED TO EXTEND
5865                                                          ;SIGN
5866  024300  010703                  MOV     PC,R3
5867  024302  000402                  BR      .+6             ;PRESERVE 2 WORDS FOR DATA
5868  024304  000000          SXRA:   .WORD   0               ;RESERVED WORD FOR DATA
5869  024306  000000          SXRB:   .WORD   0               ;RESERVED WORD FOR DATA
5870  024310  005723                  TST     (R3)+
5871  024312  010304                  MOV     R3,R4           ;R3 = ADDRESS OF SXRA
5872  024314  000250                  CLN                     ;CLEAR N BIT
5873  024316  006724                  SXT     (R4)+           ;EXTEND ZEROS INTO SXRA
5874  024320  001401                  BEQ     .+4
5875  024322  104400          SXT2:   HLT                     ;ERROR!  SXT FAILED
5876
5877  024324  010467  177754          MOV     R4,SXRA         ;SXRA = ADDRESS OF SXRB
5878  024330  000257                  CCC                     ;CLEAR CONDITION CODES
5879  024332  006733                  SXT     @(R3)+          ;EXTEND ZEROS INTO SXRB
5880  024334  001401                  BEQ     .+4
5881  024336  104400          SXT3:   HLT                     ;ERROR!
5882
5883  024340  000270                  SEN                     ;SET N BIT
5884  024342  006753                  SXT     @-(R3)          ;EXTEND ONES INTO SXRB
5885  024344  100401                  BMI     .+4
5886  024346  104400          SXT5:   HLT                     ;ERROR!
5887
5888  024350  012704  025252          MOV     #025252,R4      ;R4 = 025252
5889  024354  074433                  XOR     R4,@(R3)+       ;SXRB = 152525 (COMPLEMENT OF R4)
5890  024356  005002                  CLR     R2
5891  024360  074253                  XOR     R2,@-(R3)       ;SXRB REMAINS UNCHANGED
5892  024362  001405                  BEQ     XOR35           ;CHECK CONDITION CODES
5893  024364  100004                  BPL     XOR35
5894  024366  005104                  COM     R4              ;R4 = 152525
5895  024370  020467  177712          CMP     R4,SXRB         ;CHECK XOR
5896  024374  001401                  BEQ     .+4
5897  024376  104400          XOR35:  HLT                     ;ERROR!  XOR FAILED
5898
5899  024400  005743                  TST     -(R3)           ;R3 = ADDRESS OF SXRA-2
5900  024402  000250                  CLN                     ;CLEAR N BIT
5901  024404  006773  000002          SXT     @2(R3)          ;SXRB = 0
5902  024410  001401                  BEQ     .+4
5903  024412  104400          SXT7:   HLT                     ;ERROR!  SXT FAILED
5904
5905  024414  074473  000002          XOR     R4,@2(R3)       ;SXRB = R4
```

```
5906  024420  020473  000002          CMP     R4,@2(R3)       ;CHECK XOR
5907  024424  001401                  BEQ     .+4
5908  024426  104400          XOR7:   HLT                     ;ERROR!  XOR FAILED
5909  024430  104000                  SCOPE
5910
5911                          ;NOTE:   DO NOT INSERT ANY CODE IN FOLLOWING SOB TESTS
5912                          ;        SINCE IT TESTS THE MAXIMUM BRANCH WIDTH OF THE INSTRUCTION.
5913
5914  024432  005005                  CLR     R5              ;CLEAR ERROR INDICATOR
5915  024434  000407                  BR      SOB0            ;BRANCH TO SOB TEST
5916
5917  024436  005004          SOB10:  CLR     R4              ;R4 = 0
5918  024440  005705                  TST     R5              ;CHECK ERROR INDICATOR
5919  024442  001401                  BEQ     .+4             ;SOB BRANCHED CORRECTLY
5920  024444  104400                  HLT                     ;ERROR!
5921
5922  024446  005005          SOB9:   CLR     R5              ;CLEAR INDICATOR (R5)
5923  024450  006004                  ROR     R4              ;ROTATE RIGHT R4
5924  024452  000467                  BR      SOB8
5925
5926  024454  012700  000010  SOB0:   MOV     #10,R0          ;R0=10
5927  024460  000277                  SCC                     ;SET CONDITION CODES
5928  024462  001012          SOB1:   BNE     SOB2            ;CHECK CONDITION CODES AFTER SOB
5929  024464  100011                  BPL     SOB.            ;SOB SHOULD NOT EFFECT THE
5930  024466  102010                  BVC     SOB.            ;CONDITION CODES.
5931  024470  103007                  BCC     SOB2
5932  024472  077005                  SOB     R0,SOB1
5933  024474  001005                  BNE     SOB2            ;CHECK CONDITION CODES AFTER
5934  024476  100004                  BPL     SOB2            ;SOB FALLS THROUGH,
5935  024500  102003                  BVC     SOB2            ;SOB SHOULD NOT EFFECT
5936  024502  103002                  BCC     SOB2            ;CONDITION CODES.
5937  024504  005700                  TST     R0              ;CHECK IF R0=0
5938  024506  001401                  BEQ     .+4
5939  024510  104400          SOB2:   HLT                     ;ERROR!
5940
5941  024512  012702  000100          MOV     #100,R2         ;R2=100
5942  024516  012700  000101          MOV     #101,R0         ;SET CHECK REGISTER, R0=101
5943  024522  001414          SOB3:   BEQ     SOB4            ;CHECK CONDITION CODES AFTER
5944  024524  100413                  BMI     SOB4            ;SOB BRANCH,
5945  024526  102412                  BVS     SOB4            ;SOB SHOULD NOT EFFECT
5946  024530  103411                  BCS     SOB4            ;CONDITION CODES.
5947  024532  005300                  DEC     R0              ;DECREMENT CHECK REGISTER
5948  024534  020002                  CMP     R0,R2           ;CHECK THAT SOB DECREMENTS
5949  024536  001006                  BNE     SOB4
5950  024540  000257                  CCC                     ;SET CONDITION CODES BEFORE SOB
5951  024542  077211                  SOB     R2,SOB3         ;BRANCH TO SOB3 UNTIL R2=0
5952  024544  001403                  BEQ     SOB4            ;CHECK CONDITION CODES AFTER
5953  024546  100402                  BMI     SOB4            ;SOB FALLS THROUGH
5954  024550  005702                  TST     R2              ;CHECK IF R2=0
5955  024552  001401                  BEQ     .+4
5956  024554  104400          SOB4:   HLT                     ;ERROR!
5957
5958  024556  012700  000001  SOB5:   MOV     #1,R0           ;R0=1
5959  024562  000401                  BR      .+4
5960  024564  104400                  HLT                     ;ERROR!
5961  024566  077002                  SOB     R0,.-2          ;SOB SHOULD NOT BRANCH
```

```
5962
5963  024570  005700                    TST     R0              ;CHECK IF R0=0 AFTER SOB
5964  024572  001401                    BEQ     .+4
5965  024574  104400                    HLT                     ;ERROR!
5966
5967  024576  012704  100000  SOB5A:    MOV     #100000,R4      ;R4=100000
5968  024602  000403                    BR      1$
5969  024604  005204            3$:     INC     R4              ;R4=100000
5970  024606  100403                    BMI     2$              ;N BIT SHOULD BE SET
5971  024610  104400                    HLT                     ;ERROR!   SOB DID NOT
5972                                                            ;INCREMENT PROPERLY
5973
5974  024612  077404            1$:     SOB     R4,3$           ;SOB SHOULD BRANCH
5975  024614  104400                    HLT                     ;ERROR!  SOB DID NOT BRANCH
5976
5977  024616  012703  000100    2$:     MOV     #100,R3         ;R3=100
5978  024622  077301            SOB6:   SOB     R3,SOB6         ;USE SOB TO BRANCH TO ITSELF
5979  024624  005703                    TST     R3              ;CHECK IF R3=0
5980  024626  001703                    BEQ     SOB10
5981  024630  104400            SOB7:   HLT                     ;ERROR!
5982
5983  024632  005705            SOB8:   TST     R5              ;CHECK INDICATOR (R5)
5984                                                            ;IF SOB BRANCHES INCORRECTLY
5985                                                            ;WHEN CHECKING MAX. BRANCH,
5986                                                            ;R5 WILL NOT BE CLEARED AT
5987                                                            ;THIS POINT INDICATING AN ERROR.
5988
5989  024634  001401                    BEQ     .+4             ;BRANCH IF SOB BRANCHES CORRECTLY
5990  024636  104400                    HLT                     ;ERROR!
5991
5992  024640  005205                    INC     R5              ;SET INDICATOR (R5)
5993  024642  077477                    SOB     R4,SOB9         ;TEST MAX. BRANCH OF SOB
5994  024644  005704                    TST     R4              ;CHECK IF R4=0
5995  024646  001401                    BEQ     .+4
5996  024650  104400                    HLT                     ;ERROR!
5997  024652  104000                    SCOPE
5998
5999
6000                                    ;CHECK THAT MARK INSTRUCTION POPS OVER THE CORRECT # OF ARGUMENTS,RESTORES R5 FR
6001                                    ;THE STACK POINTER
6002
6003  024654  010602          MRKTST:   MOV     SP,R2
6004  024656  010705                    MOV     PC,R5           ;THE STACK LOOKS LIKE THIS AFTER
6005  024660  010500                    MOV     R5,R0           ;THE JSR INSTRUCTION
6006  024662  010546                    MOV     R5,-(SP)        ; -2(SP)= R0    THIS IS A
6007  024664  010746                    MOV     PC,-(SP)        ; -4(SP)= PC    STRING
6008  024666  010746                    MOV     PC,-(SP)        ; -6(SP)= PC+2  OF
6009  024670  010746                    MOV     PC,-(SP)        ;-10(SP)= PC+4  FIVE
6010  024672  010746                    MOV     PC,-(SP)        ;-12(SP)= PC+6  DUMMY
6011  024674  010746                    MOV     PC,-(SP)        ;-14(SP)= PC+10 ARGUMENTS
6012  024676  012746  006405            MOV     #MARK+5,-(SP)   ;-16(SP)= MARK 5
6013  024702  010605                    MOV     SP,R5           ;-20(SP)= PC PUSHED BY JSR
6014  024704  004767  000002            JSR     PC,MARK1
6015  024710  000403                    BR      .+10
6016  024712  000205          MARK1:    RTS     R5
6017  024714  104400                    HLT                     ;ERROR! SHOULD BE DOING MARK 5 INST.
```

```
 6018  024716  000407                        BR      MARKEX
 6019  024720  020602                        CMP     SP,R2
 6020  024722  001402                        BEQ     .+6
 6021  024724  104400                        HLT                     ;ERROR! SP NOT RETURNED TO PROPER
 6022  024726  000403                        BR      MARKEX          ;VALUE BY MARK INSTRUCTION
 6023  024730  020005                        CMP     R0,R5
 6024  024732  001401                        BEQ     .+4
 6025  024734  104400                        HLT                     ;ERROR! DID NOT RESTORE R5 FROM STACK
 6026  024736  010206              MARKEX:   MOV     R2,SP           ;RESTORE SP
 6027  024740  104000                        SCOPE
 6028
 6029                                ;RTT/RTI TEST INSURES THAT CP DOES THE INSTRUCTION FOLLOWING
 6030                                ;AN RTT IF THE "T"BIT IS SET IN THE PSW,BUT DOES HONOR
 6031                                ;THE TRAP IMMEDIATELY IF IT EXECUTES AN RTI
 6032                                ;INSTRUCTION SEQUENCE-RTT
 6033                                ;2$:    RTT                     ;NO 'T' TRAP AFTER RTT
 6034                                ;       INC     R0              ;R0=000001
 6035                                ;                               ;'T' TRAP TO 5$ AFTER INC
 6036                                ;5$:    COM     R0              ;R0=177776
 6037                                ;       MOV     SAVPSW,2(SP)    ;CLEAR 'T' BIT IN RETURN PSW
 6038                                ;       RTI                     ;RETURN TO INSTRUCTION FOLLOWING INC
 6039                                ;       CMP     #RTT,2$         ;CHECK
 6040                                ;       ETC
 6041
 6042                                ;INSTRUCTION SQUENCE-RTI
 6043                                ;2$:    RTI                     ;'T' TRAP AFTER RTI
 6044                                ;5$:    COM     R0              ;R0=177777
 6045                                ;       MOV     SAVPSW,2(SP)    ;CLEAR 'T' BIT IN RETURN PSW
 6046                                ;       RTI                     ;RETURN TO INC INSTRUCTION
 6047                                ;       INC     R0              ;R0=000000
 6048                                ;       CMP     #RTT,2$         ;CHECK
 6049                                ;       ETC
 6050  024742  013767  177776  000166 RTT1:   MOV     @#PSW,SAVPSW            ;SAVE PSW
 6051  024750  032767  000020  000160        BIT     #T,SAVPSW               ;CHECK IF "T"BIT SET
 6052  024756  001176                        BNE     RTT2EX                  ;BRANCH TO EXIT
 6053  024760  010746                1$:     MOV     PC,-(SP)                ;GET CURRENT PC
 6054  024762  062716  000116                ADD     #5$-.,(SP)              ;FORM RELOCATED PC
 6055  024766  012637  000014                MOV     (SP)+,@#TBITVEC         ;LOAD INTO TRAP VECTOR
 6056  024772  016746  000140                MOV     SAVPSW,-(SP)            ;GET CURRENT PSW
```

```
 6057  024776  011637  000016                MOV     (SP),@#TBITVEC+2
 6058  025002  052737  000340  177776        BIS     #PRTY7,@#PSW            ;SET PRIORITY LEVEL 7
```

```
6059  025010  005000                          CLR     R0
6060  025012  052716  000360                  BIS     #PRTY7+T,(SP)                   ;SET "T"BIT IN PSW ON STACK
6061  025016  010746                          MOV     PC,-(SP)                ;PUT THE PC ON THE STACK
6062  025020  062716  000006                  ADD     #6,(SP)                 ;ADJUST PC FOR NEXT INSTRUCTION
6063  025024  000006                  2$:     RTT
6064  025026  005200                          INC     R0                      ;DONE TO SEE IF INSTR. FOLLOWING
6065                                                                           ;RTT IS EXECUTED IF T-BIT SET
6066  025030  042737  000340  177776          BIC     #PRTY7,@#PSW            ;SET PRIORITY LEVEL 0
6067  025036  022767  000006  177760          CMP     #RTT,2$
6068  025044  001005                          BNE     3$                      ;CHECK IF INC WAS EXECUTED
6069  025046  022700  177776                  CMP     #177776,R0              ;CHECK IF COM-R0 EXECUTED
6070  025052  001406                          BEQ     4$
6071  025054  104400                          HLT                             ;ERROR!R0 NOT COMPLIMENTED
6072  025056  000415                          BR      6$                      ;EXIT TEST
6073  025060  005700                  3$:     TST     R0                      ;TEST IF TRAPED BEFORE INC INST.
6074                                                                           ;WAS EXECUTED
6075  025062  001413                          BEQ     6$
6076  025064  104400                          HLT                             ;ERROR!
6077  025066  000411                          BR      6$                      ;EXIT TEST
6078  025070  012767  000002  177726  4$:     MOV     #RTI,2$
6079  025076  000730                          BR      1$
6080  025100  005100                  5$:     COM     R0                      ;RTT CHECK
6081  025102  016766  000030  000002          MOV     SAVPSW,2(SP)
6082  025110  000002                          RTI
6083  025112  012767  000006  177704  6$:     MOV     #RTT,2$
6084  025120  012737  000016  000014          MOV     #TBITVEC+2,@#TBITVEC    ;RESTORE TRAP VECTORES
6085  025126  005037  000016                  CLR     @#TBITVEC+2
6086  025132  104000                  RTT1EX: SCOPE
6087
6088  025134  000401                          BR      RTT2
6089  025136  000000                  SAVPSW: .WORD   0
6090                                  ;CHECK IF AN 11/45 AND DETERMINE WHICH MODE AND REG. SET ARE SELECTED BY THE PSW
6091  025140  122737  000004  000766  RTT2:   CMPB    #4,@#OPT.CP             ;TEST IF AN 11/40
6092  025146  001002                          BNE     RTT2A                   ;BRANCH IF NOT AN 11/40
6093  025150  000167  000200                  JMP     RTT2EX                  ;GO TO RTT2EX IF 11/40
6094  025154  016700  177756          RTT2A:  MOV     SAVPSW,R0               ;GET SAVED PSW
6095  025160  105000                          CLRB    R0                      ;CLEAR PRIOITY LEVEL,T, AND COND CODES
6096  025162  012702  144000                  MOV     #UM+REG,R2
6097  025166  074002                          XOR     R0,R2
6098  025170  001435                          BEQ     2$                      ;USER MODE REG. SET #1 ON
6099  025172  012702  044000                  MOV     #SM+REG,R2
6100  025176  074002                          XOR     R0,R2
6101  025200  001447                          BEQ     3$                      ;SUPER MODE REG. SET #1 ON
6102  025202  032700  140000                  BIT     #UM,R0
6103  025206  001062                          BNE     RTT2EX
6104
6105                                  ;TEST THAT RTT CLEARS BITS 11,12,13 & PRIORITY LEVEL BITS IN KERNEL MODE
6106  025210  012702  177777                  MOV     #-1,R2                  ;KERNEL MODE REG. SET 0 ON
6107  025214  012737  034240  177776          MOV     #PUM+REG+PRTY5,@#PSW    ;SELECT REG. SET #1
6108  025222  005002                          CLR     R12                     ;SHOULD CLEAR REG #12
6109  025224  012746  000100                  MOV     #PRTY2,-(SP)
6110  025230  010746                          MOV     PC,-(SP)
6111  025232  062716  000006                  ADD     #1$-.,(SP)              ;FORM NEW PC
6112  025236  000006                          RTT
6113  025240  013700  177776          1$:     MOV     @#PSW,R0                ;NOW USING REG SET 0
6114  025244  005702                          TST     R2                      ;SHOULD TEST R2 NOT R12
```

```
6115  025246  001001                          BNE     4$
6116  025250  104400                          HLT                             ;ERROR!DID NOT CLEAR BIT #11 OF PSW
6117  025252  022700  000100          4$:     CMP     #PRTY2,R0               ;TESTS THE PSW AFTER THE RTT
6118  025256  001436                          BEQ     RTT2EX
6119  025260  104400                          HLT                             ;ERROR! INCORRECT PSW AFTER THE RTT
6120  025262  000434                          BR      RTT2EX
6121
6122                                  ;TEST TO INSURE THAT RTI DOES NOT CLEAR BITS 11-15 IN USER MODE
6123  025264  052737  030340  177776  2$:     BIS     #PUM+PRTY7,@#PSW        ;PSW<15-5>=144X
6124  025272  005046                          CLR     -(SP)
6125  025274  010746                          MOV     PC,-(SP)
6126  025276  062716  000006                  ADD     #5$-.,(SP)
6127  025302  000002                          RTI                             ;ATTEMPS TO INSERT A PSW OF 0
6128  025304  022737  174340  177776  5$:     CMP     #UM+PUM+REG+PRTY7,@#PSW ;SHOULD CHECK AGAINST REG #0
6129  025312  001420                          BEQ     RTT2EX
6130  025314  104400                          HLT                             ;ERROR! RTI CLEARED BITS IN PSW
6131  025316  000416                          BR      RTT2EX
6132
6133                                  ;TEST THAT BITS 11-15 AND PRIORTY BITS ARE NOT ALTERED IN SUPER MODE
6134  025320  052737  030200  177776  3$:     BIS     #PUM+PRTY4,@#PSW        ;PSW<15-5>=044X
6135  025326  012746  000340                  MOV     #PRTY7,-(SP)
6136  025332  010746                          MOV     PC,-(SP)
6137  025334  062716  000006                  ADD     #6$-.,(SP)
6138  025340  000006                          RTT                             ;ATTEMPTS TO CLEAR 11-15 AND ALTER PRTY
6139
6140  025342  022737  074200  177776  6$:     CMP     #SM+PUM+REG+PRTY4,@#PSW
6141  025350  001401                          BEQ     RTT2EX
6142  025352  104400                          HLT                             ;ERROR! RTT ALTERED PRTY IN
6143                                                                          ;SUPER MODE OR BITS 11-15.
6144  025354  016737  177556  177776  RTT2EX: MOV     SAVPSW,@#PSW
6145  025362  104000                          SCOPE
6146
6147  025364  010702                          MOV     PC,R2
6148  025366  062702  000012                  ADD     #12,R2
6149  025372  012707  001152                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
6150  025376  000000                  REL55:  .WORD   0
6151                                  ;5555555555555 LAST ADDRESS OF CODE TO BE RELOCATED 55555555555
6152
6153
6154
6155                                          .SBTTL  START OF SECTION 6
6156                                  ;6666666666666 FIRST ADDRESS TO BE RELOCATED 666666666
6157  025400  010700                  REL6:   MOV     PC,R0                   ;GET PC
6158  025402  005740                          TST     -(R0)                   ;R0 CONTAINS THE ADDRESS OF REL6
6159  025404  010037  001010                  MOV     R0,@#FRSTAD             ;SAVE
6160  025410  012737  000006  005442          MOV     #6,@#SECT               ;SET SECTION #
6161  025416  004737  005432                  JSR     PC,@#LDDISP             ;LOAD DISPLAY GEG
6162  025422  013767  005436  001712          MOV     @#DISPLY,REL66
6163  025430  010700                          MOV     PC,R0                   ;GET CURRENT PC
6164  025432  162700  025432                  SUB     #.,R0                   ;SUBTRACT RELOCATION FACTOR
6165  025436  010037  001004                  MOV     R0,@#FACTOR             ;SAVE RELOCATION FACTOR
6166  025442  010701                          MOV     PC,R1                   ;SET NEW SCOPE PTR
6167
6168  025444  032737  040000  000766          BIT     #EISOPT,@#OPT.CP        ;CHECK IF 11/40 WITH EIS OPTION
6169  025452  001002                          BNE     ASHLO                   ;BRANCH IF EIS OPT AVAIL.
6170  025454  000167  001340                  JMP     MPI
```

```
 6171
 6172                                        ;CHECK ASH,ASHC,MUL, AND DIV INSTRUCTIONS
 6173  025460  012700  000001        ASHLO:  MOV     #1,R0           ;R0 WILL BE THE SHIFT COUNT
 6174  025464  012703  000021                MOV     #17.,R3         ;MAX SHIFT COUNT
 6175  025470  005067  000014        1$:     CLR     2$              ;PRESET SAVED CC'S LOCATION=0
 6176  025474  010002                        MOV     R0,R2           ;GET SHIFT COUNT FOR PASS
 6177  025476  010705                        MOV     PC,R5           ;R5 & R4 WILL BE DATA SHIFTED BY
 6178  025500  010504                        MOV     R5,R4           ;ASH & ASL INSTRUCTIONS
 6179  025502  072502                        ASH     R2,R5           ;SHIFT R5
 6180  025504  113727  177776                MOVB    @#PSW,(PC)+     ;SAVE CC'S
 6181  025510  000000        2$:     .WORD   0                       ;CONTAINS ASH CC'S IN EVEN BYTE
 6182                                                                ;ASL CC'S IN ODD BYTE
 6183  025512  006304        3$:     ASL     R4                      ;SHIFT R4
 6184  025514  113746  177776                MOVB    @#PSW,-(SP)     ;SAVE PSW ON STACK
 6185  025520  132716  000002                BITB    #V,(SP)         ;CHECK IF ASL SET V BIT
 6186  025524  00.403                         BEQ     30$
 6187  025526  152767  000002  177755        BISB    #V,2$+1         ;IF ASL SET V THEN SET V IN 2$+1
 6188  025534  112637  177776        30$:    MOVB    (SP)+,@#PSW     ;RESTORE ORIGINAL PSW
 6189  025540  077214                        SOB     R2,3$           ;SHIFT R4 R2 TIMES
 6190  025542  153767  177776  177741        BISB    @#PSW,2$+1      ;SAVE CC'S AFTER ASL
 6191  025550  020504                        CMP     R5,R4           ;CHECK ASH & ASL RESULTS
 6192  025552  001004                        BNE     4$
 6193  025554  126767  177730  177727        CMPB    2$,2$+1         ;CHECK ASH & ASL CC'S
 6194  025562  001401                        BEQ     .+4
 6195  025564  104400        4$:     HLT                             ;ERROR! INCORRECT RESULT OR CC'S
 6196  025566  005200                        INC     R0              ;INCREMENT PASS SHIFT COUNT
 6197  025570  020003                        CMP     R0,R3
 6198  025572  001336                        BNE     1$
 6199
 6200  025574  012700  177777        ASHRO:  MOV     #-1,R0          ;R0 = RIGHT SHIFT COUNT FOR PASS
 6201  025600  012703  177757                MOV     #-17.,R3        ;MAX SHIFT COUNT
 6202  025604  010002        1$:     MOV     R0,R2           ;GET SHIFT COUNT FOR PASS
 6203  025606  010705                        MOV     PC,R5           ;R5 & R4 = DATA TO BE SHIFTED
 6204  025610  010504                        MOV     R5,R4           ;BY ASH & ASR INSTRUCTIONS
 6205  025612  072502                        ASH     R2,R5           ;SHIFT R5 R2 TIMES
 6206  025614  113727  177776                MOVB    @#PSW,(PC)+     ;SAVE CC'S IN EVEN BYTE
 6207  025620  000000        2$:     .WORD   0                       ;CONTAINS ASH CC'S IN EVEN BYTE
 6208                                                                ;ASR CC'S IN ODD BYTE
 6209  025622  005402                        NEG     R2
 6210  025624  006204        3$:     ASR     R4                      ;SHIFT R4
 6211  025626  077202                        SOB     R2,3$           ;SHIFT R4 R2 TIMES
 6212  025630  113767  177776  177763        MOVB    @#PSW,2$+1      ;SAVE CC'S AFTER ASR
 6213  025636  142767  000002  177755        BICB    #V,2$+1         ;ASH RIGHT WILL NOT SET V ASR MAY SET V
 6214  025644  020504                        CMP     R5,R4           ;CHECK ASH & ASR RESULTS
 6215  025646  001004                        BNE     4$
 6216  025650  126767  177744  177743        CMPB    2$,2$+1         ;CHECK ASH & ASR CC'S
 6217  025656  001401                        BEQ     .+4
 6218  025660  104400        4$:     HLT
 6219  025662  005300                        DEC     R0              ;DECREMENT PASS SHIFT COUNT
 6220  025664  020003                        CMP     R0,R3
 6221  025666  001346                        BNE     1$
 6222
 6223  025670  012746  000037        ASHCLO: MOV     #31.,-(SP)      ;PUT MAX SHIFT COUNT ON STACK
 6224  025674  012746  000001                MOV     #1,-(SP)        ;PUT LEFT SHIFT COUNT ON STACK
 6225  025700  011600        1$:     MOV     (SP),R0         ;GET PASS SHIFT COUNT
 6226  025702  010705                        MOV     PC,R5           ;CURRENT PC IS DATA TO BE SHIFTED
```

```
6227  025704  010503                    MOV     R5,R3           ;ASHC SHIFTS R4,R5;ASL,ROL SHIFTS R2,R3
6228  025706  005004                    CLR     R4
6229  025710  005002                    CLR     R2
6230  025712  073400                    ASHC    R0,R4           ;SHIFT R4 LEFT AS SPECIFIED BY R0
6231  025714  006303            2$:     ASL     R3              ;SHIFT R2,R3 LEFT
6232  025716  006102                    ROL     R2              ;AS SPECIFIED BY R0
6233  025720  077003                    SOB     R0,2$
6234  025722  020402                    CMP     R4,R2           ;CHECK RESULTS
6235  025724  001002                    BNE     3$
6236  025726  020503                    CMP     R5,R3
6237  025730  001401                    BEQ     .+4
6238  025732  104400            3$:     HLT
6239  025734  005216                    INC     (SP)            ;INCREMENT NEXT PASS SHIFT COUNT
6240  025736  021666  000002            CMP     (SP),2(SP)      ;REACHED MAX COUNT (31.)
6241  025742  001356                    BNE     1$
6242  025744  022626                    CMP     (SP)+,(SP)+     ;RESTORE STACK PTR
6243
6244  025746  012746  177740  ASHCR0:   MOV     #-32.,-(SP)     ;PUT MAX RIGHT SHIFT COUNT ON STACK
6245  025752  012746  177777            MOV     #-1,-(SP)       ;PUT PASS SHIFT COUNT ON STACK
6246  025756  011600            1$:     MOV     (SP),R0         ;GET PASS SHIFT COUNT
6247  025760  010702                    MOV     PC,R2           ;R2,R3 & R4,R5 ARE THE DATA REGISTERS
6248  025762  010204                    MOV     R2,R4           ;TO BE SHIFTED BY TEST
6249  025764  005003                    CLR     R3
6250  025766  005005                    CLR     R5
6251  025770  000262                    SEV                     ;SET V BIT IN PSW
6252  025772  073200                    ASHC    R0,R2           ;SHIFT R2,R3 RIGHT R0 TIMES
6253  025774  102410                    BVS     3$              ;SHIFT RIGHT CLEARS V
6254  025776  005400                    NEG     R0              ;NEGATE SHIFT COUNT FOR SOB
6255  026000  006204            2$:     ASR     R4              ;SHIFT R4,R5 RIGHT R0 TIMES
6256  026002  006005                    ROR     R5
6257  026004  077003                    SOB     R0,2$
6258  026006  020204                    CMP     R2,R4           ;CHECK RESULT
6259  026010  001002                    BNE     3$
6260  026012  020305                    CMP     R3,R5
6261  026014  001401                    BEQ     .+4
6262  026016  104400            3$:     HLT
6263  026020  005316                    DEC     (SP)            ;SET SHIFT COUNT FOR NEXT PASS
6264  026022  021666  000002            CMP     (SP),2(SP)      ;CHECK IF MAX SHIFT COUNT
6265  026026  001353                    BNE     1$
6266  026030  022626                    CMP     (SP)+,(SP)+     ;RESTORE STACK PTR
6267  026032  104000                    SCOPE
6268
6269                              ;THE BELOW TEST OF THE MUL INSTRUCTION MULTIPLIES THE CURRENT PC
6270                              ;BY 1,2,4,8 ETC AND SHIFTS THE SAME PC VALUE USING AN ASHC LEFT BY
6271                              ;0,1,2,3,ETC AND COMPARES THE RESULTS. CONDITION CODE RESULTS ARE NOT CHECKED.
6272  026034  012700  000001    MULO:   MOV     #1,R0           ;R0 CONTAINS MULTIPLIER FOR MUL
6273  026040  005016                    CLR     (SP)            ;(SP) CONTAINS SHIFT VALUE FOR ASHC
6274  026042  010702            1$:     MOV     PC,R2           ;R3,R2 & R5,R4 ARE DATA REGISTERS
6275  026044  010227                    MOV     R2,(PC)+        ;SAVE MULTIPICAND
6276  026046  000000                    .WORD   0               ;CONTAINS ORIGINAL MULTIPICAND
6277  026050  005003                    CLR     R3
6278  026052  005004                    CLR     R4
6279  026054  010205                    MOV     R2,R5           ;FOR MUL AND ASHC
6280  026056  100001                    BPL     .+4             ;IF MULTPICAND IS NEG THEN SET R4 = -1
6281  026060  005104                    COM     R4              :FOR ASHC
6282  026062  000277                    SCC                     ;PRESET CC'S
```

```
 6283  026064  070200                    MUL     R0,R2           ;MULTIPLY R2 BY R0 LEAVE PRODUCT
 6284                                                            ;IN R2,R3 MSH IN R2,LSH IN R3
 6285  026066  102406                    BVS     2$
 6286  026070  001405                    BEQ     2$              ;PRODUCT WILL NEVER BE = 0
 6287  026072  073416                    ASHC    (SP),R4         ;'MULTIPLY' R4,R5 BY (SP) LEAVE PRODUCT
 6288                                                            ;IN R4,R5 MSH IN R4,LSH IN R5
 6289  026074  020204                    CMP     R2,R4           ;CHECK MSH RESULT
 6290  026076  001002                    BNE     2$
 6291  026100  020305                    CMP     R3,R5           ;CHECK LSH RESULT
 6292  026102  001401                    BEQ     .+4
 6293  026104  104400            2$:     HLT
 6294  026106  005216                    INC     (SP)            ;INCREMENT ASHC SHIFT COUNT
 6295  026110  006300                    ASL     R0              ;SHIFT MUL MULTIPLIER
 6296  026112  102353                    BVC     1$
 6297                            ;CHECK MUL INST WITH MULTIPIER (R0) = 100000
 6298  026114  010702                    MOV     PC,R2           ;R2 = MULTIPICAND
 6299  026116  005202                    INC     R2
 6300  026120  010227                    MOV     R2,(PC)+        ;SAVE MULTIPICAND
 6301  026122  000000                    .WORD   0               ;CONTAINS ORIGINAL MULTIPICAND
 6302  026124  005103                    COM     R3
 6303  026126  010204                    MOV     R2,R4           ;R4 WILL BE MSH 'PRODUCT'
 6304  026130  006204                    ASR     R4              ;FORM 'PRODUCT'
 6305  026132  005104                    COM     R4              ;COMPLEMENT MSH 'PRODUCT'
 6306  026134  070200                    MUL     R0,R2           ;MULTIPLY R2 BY 100000 LEAVING
 6307                                                            ;R2 = MSH, R3 = LSH PRODUCT
 6308  026136  020204                    CMP     R2,R4           ;COMPARE MSH PRODUCTS
 6309  026140  001002                    BNE     3$
 6310  026142  020003                    CMP     R0,R3           ;CHECK LSH PRODUCT
 6311  026144  001401                    BEQ     .+4
 6312  026146  104400            3$:     HLT
 6313  026150  104000                    SCOPE
 6314
 6315                            ;THE BELOW TEST OF THE DIV INSTRUCTION DIVIDES THE CURRENT PC BY
 6316                            ;1,2,4,8,ETC LEAVING THE QUOTIENT/REMAINDER IN R2/R3. NEXT THE QUOTIENT
 6317                            ;IS MULTPLIED BY 1,2,4,8,ETC AND THE REMAINDER ADDED. THE RESULT IS
 6318                            ;THEN COMPARED WITH THE ORIGINAL CURRENT PC.
 6319  026152  012700  000001    DIV0:   MOV     #1,R0           ;R0=DIVISOR
 6320  026156  010737  026250            MOV     PC,@#10$        ;SAVE DATA IN 10$
 6321  026162  013703  026250    1$:     MOV     @#10$,R3        ;GET DATA
 6322  026166  005002                    CLR     R2              ;CLEAR MSH DIVIDEND
 6323  026170  000277                    SCC
 6324  026172  071200                    DIV     R0,R2           ;DIVIDE R2 BY R0 LEAVING QUOTIENT IN R2
 6325                                                            ;AND REMAINDER IN R3
 6326  026174  103421                    BCS     2$
 6327  026176  100420                    BMI     2$
 6328  026200  102010                    BVC     20$             ;BRANCH IF DIVIDE WORKED
 6329  026202  022700  000001            CMP     #1,R0           ;V BIT SHOULD ONLY SET IF DIVIDING BY 1
 6330  026206  001014                    BNE     2$              ;AND THE LSH OF DIVIDEND
 6331  026210  032737  100000  026250    BIT     #100000,@#10$   ;IS NEGATIVE
 6332  026216  001410                    BEQ     2$
 6333  026220  000410                    BR      3$
 6334  026222  010204            20$:    MOV     R2,R4           ;GET QUOTIENT
 6335  026224  070400                    MUL     R0,R4           ;MULTIPLY QUOTIENT BY DIVISOR
 6336  026226  060305                    ADD     R3,R5           ;ADD REMAINDER TO LSH PRODUCT
 6337  026230  103403                    BCS     2$              ;SHOULD BE NO CARRY
 6338  026232  023705  026250            CMP     @#10$,R5        ;CHECK RESULT
```

```
 6339  026236  001401                        BEQ     .+4
 6340  026240  104400                2$:     HLT                     ;ERROR! DIVIDE FAILED
 6341                                                                  ;QUOTIENT IS IN R2,REMAINDER IN R3
 6342                                                                  ;ORIGINAL PC IS IN 10$ AND FINAL
 6343                                                                  ;PRODUCT IN R4,R5 [MSH][LSH]
 6344  026242  006300                3$:     ASL     R0              ;GET NEXT DIVISOR
 6345  026244  102346                        BVC     1$
 6346  026246  000401                        BR      ASHL1
 6347  026250  000000                10$:    .WORD   0               ;CONTAINS ORIGINAL PC USED AS DATA
 6348
 6349                                ;CHECK ASH,ASHC,MUL, AND DIV INSTRUCTIONS USING ADDRESS MODE 1
 6350  026252  005016                ASHL1:  CLR     (SP)            ;(SP) = SHIFT COUNT
 6351  026254  005000                        CLR     R0              ;R0 = SHIFT COUNT FOR CHECK ASH
 6352  026256  012702  000020                MOV     #16.,R2         ;R2 = MAX LEFT SHIFT COUNT
 6353  026262  005067  000012        1$:     CLR     2$              ;CLEAR CC'S HOLDING ADDRESS
 6354  026266  010703                        MOV     PC,R3           ;R3,R4 = DATA TO BE SHIFTED
 6355  026270  010304                        MOV     R3,R4
 6356  026272  072316                        ASH     (SP),R3         ;SHIFT R3 LEFT (SP) TIMES
 6357  026274  013727  177776                MOV     @#PSW,(PC)+     ;SAVE CC'S
 6358  026300  000000                2$:     .WORD   0               ;CONTAINS ASH (SP),R3 CC'S IN EVEN BYTE
 6359                                                                  ;AND ASH R0,R4 CC'S IN ODD BYTE
 6360  026302  072400                        ASH     R0,R4           ;SHIFT R4 LEFT R0 TIMES
 6361  026304  113767  177776  177767        MOVB    @#PSW,2$+1      ;SAVE CC'S IN ODD BYTE OF 2$
 6362  026312  020304                        CMP     R3,R4           ;COMPARE RESULTS
 6363  026314  001004                        BNE     3$              ;BRANCH IF THEY DO NOT COMPARE
 6364  026316  126767  177756  177755        CMPB    2$,2$+1         ;CHECK CC'S AFTER ASH INSTRUCTIONS
 6365  026324  001401                        BEQ     .+4
 6366  026326  104400                3$:     HLT                     ;ERROR! EITHER RESULTS OF SHIFT OR
 6367                                                                  ;RESULT CC'S ARE INCORRECT
 6368  026330  005200                        INC     R0              ;INCREMENT SHIFT COUNT FOR ASH R0,R4
 6369  026332  005216                        INC     (SP)            ;INCRENENT SHIFT COUNT FOR ASH (SP),R3
 6370  026334  020200                        CMP     R2,R0           ;CHECK FOR MAX SHIFT COUNT
 6371  026336  001351                        BNE     1$
 6372
 6373  026340  005016                ASHR1:  CLR     (SP)            ;(SP) = SHIFT COUNT FOR ASH (SP),R4
 6374  026342  005000                        CLR     R0              ;R0 = SHIFT COUNT FOR ASH R0,R5
 6375  026344  005402                        NEG     R2              ;R2 = MAX RIGHT SHIFT COUNT (SET BY
 6376                                                                  ;ABOVE TEST TO 16. NOW = -16.
 6377  026346  005067  000012        1$:     CLR     2$              ;CLEAR CC'S HOLDING ADDRESS
 6378  026352  010704                        MOV     PC,R4           ;R4,R5 = DATA TO BE SHIFTED RIGHT
 6379  026354  010405                        MOV     R4,R5
 6380  026356  072416                        ASH     (SP),R4         ;SHIFT R4 RIGHT (SP) TIMES
 6381  026360  013727  177776                MOV     @#PSW,(PC)+     ;SAVE CC'S
 6382  026364  000000                2$:     .WORD   0               ;CONTAINS ASH (SP),R4 CC'S IN EVEN BYTE
 6383                                                                  ;AND ASH R0,R5 CC'S IN ODD BYTE
 6384  026366  072500                        ASH     R0,R5           ;SHIFT R5 RIGHT R0 TIMES
 6385  026370  113767  177776  177767        MOVB    @#PSW,2$+1      ;SAVE CC'S IN ODD BYTE 2$
 6386  026376  020405                        CMP     R4,R5           ;CHECK RESULTS
 6387  026400  001004                        BNE     3$
 6388  026402  126767  177756  177755        CMPB    2$,2$+1         ;CHECK RESULT CC'S
 6389  026410  001401                        BEQ     .+4
 6390  026412  104400                3$:     HLT                     ;ERROR! EITHER RESULTS OR RESULT CC'S
 6391                                                                  ;DID NOT COMPARE
 6392  026414  005300                        DEC     R0              ;DECREMENT SHIFT COUNT
 6393  026416  005316                        DEC     (SP)            ;DECREMENT SHIFT COUNT FOR ASH (SP),R4
 6394  026420  020002                        CMP     R0,R2           ;CHECK FOR MAX RIGHT SHIFT
```

```
6395  026422  001351                          BNE     1$
6396  026424  104000                          SCOPE
6397
6398  026426  122737  000004  000766          CMPB    #4,@#OPT.CP     ;CHECK IF AN 11/40
6399  026434  001002                          BNE     SPLTST          ;BRANCH IF NOT AN 11/40
6400  026436  000167  000356                  JMP     MPI             ;GO TO MPI IF 11/40
6401
6402                                  ;CHECK SPL INSTRUCTION
6403  026442  012702                  SPLTST: MOV     (PC)+,R2        ;R2 CONTAINS OP CODE FOR SPL  7
6404  026444  000237                          SPL     7
6405  026446  005004                          CLR     R4
6406  026450  042744  000340                  BIC     #PRTY7,-(R4)    ;CLEAR PRIORITY 'EVEL BITS IN PSW
6407  026454  011403                          MOV     (R4),R3         ;GET CURRENT PSW
6408  026456  042703  177757                  BIC     #177757,R3      ;R3 CONTAINS CORRECT PSW AFTER SPL
6409
6410  026462  012767  000230  000010          MOV     #SPL+0,2$       ;INITIALIZE SPL INSTRUCTIONS
6411  026470  012767  000237  000050          MOV     #SPL+7,5$
6412  026476  000257                  1$:     CCC                     ;CLEAR CONDITION CODES
6413  026500  000230                  2$:     SPL     0               ;SET PRIORITY LEVEL (NOTE: SPL=NOP IF USER/SUPER MODE)
6414  026502  121403                          CMPB    (R4),R3         ;CHECK RESULT OF SPL ABOVE
6415  026504  001401                          BEQ     .+4
6416  026506  104400                          HLT                     ;ERROR! SPL ABOVE FAILED
6417  026510  032714  140000                  BIT     #UM,(R4)        ;IF NOT IN KERNEL MODE THEN SPL
6418  026514  001002                          BNE     3$              ;ACTS AS A NOP
6419  026516  062703  000040                  ADD     #40,R3          ;SET NEXT CORRECT PSW RESULT
6420  026522  005267  177752          3$:     INC     2$              ;SET NEXT SPL INSTRUCTION
6421  026526  026702  177746                  CMP     2$,R2           ;CHECK IF DONE
6422  026532  002761                          BLT     1$              ;LOOP UNTIL DONE CHANGING SPL EACH PASS
6423  026534  012702                          MOV     (PC)+,R2        ;R2 CONTAINS SPL INSTRUCTION BELOW
6424  026536  000230                          SPL     0
6425  026540  052703  000017                  BIS     #17,R3          ;SET CONDITION CODE RESULT INTO R3
6426  026544  000277                  4$:     SCC                     ;SET CONDITION CODES
6427  026546  000237                  5$:     SPL     7               ;SET PRIORITY LEVEL
6428  026550  121403                          CMPB    (R4),R3         ;CHECK RESULT OF SPL ABOVE
6429  026552  001401                          BEQ     .+4
6430  026554  104400                          HLT                     ;ERROR! SPL ABOVE FAILED
6431  026556  032714  140000                  BIT     #UM,(R4)        ;CHECK IF IN KERNEL MODE
6432  026562  001002                          BNE     6$
6433  026564  162703  000040                  SUB     #40,R3          ;SET NEXT CORRECT PSW RESULT
6434  026570  005367  177752          6$:     DEC     5$              ;SET NEXT SPL
6435  026574  026702  177746                  CMP     5$,R2           ;CHECK IF DONE ALL SPL'S
6436  026600  002361                          BGE     4$
6437  026602  104000                          SCOPE
6438
6439                                  ;CHECK PROGRAM INTERRUPT REQUEST LOGIC
6440                                  ;THIS TEST CHECKS THAT WHEN A REQUEST IS MADE AT A LEVEL = TO THE
6441                                  ;CURRENT PROCESSER PRIORITY LEVEL THAT NO INTERRUPT TAKES PLACE, AND
6442                                  ;THAT WHEN A REQUEST IS MADE AT A LEVEL 1 GREATER THAN THE CURRENT PRO-
6443                                  ;CESSER LEVEL THAT AN INTERRUPT OCCURS
6444  026604  012700  026744          PIRQ0:  MOV     #4$,R0          ;R0 POINTS TO A TABLE OF CORRECT PIRQ
6445                                                                  ;CONTENTS AFTER AN INTERRUPT
6446  026610  012702  000400                  MOV     #400,R2         ;R2 CONTAINS INTERRUPT REQUEST LEVEL
6447  026614  005003                          CLR     R3              ;R3 CONTAINS PROCESSER PRIORITY LEVEL
6448  026616  012704  177772                  MOV     #PIRQ,R4        ;R4 CONTAINS ADDRESS OF PIRQ REGISTER
6449  026622  005014                          CLR     (R4)            ;INITIALZE REQUEST LEVEL TO 0
6450  026624  013737  177776  000242          MOV     @#PSW,@#PIRVEC+2        ;RETAIN MODE & REG SET ON TRAP
```

```
6451  026632  112737  000340  000242          MOVB    #PRTY7,@#PIRVEC+2       ;ASSUME LEVEL 7 ON INTERRUPT
6452  026640  012737  026742  000014          MOV     #20$,@#TBITVEC          ;SET NEW TBIT TRAP VECTTOR
6453  026646  012737  000340  000016          MOV     #PRTY7,@#TBITVEC+2      ;PRIORITY LEVEL 7 ON TRAP
6454  026654  012737  026714  000240  1$:     MOV     #2$,@#PIRVEC            ;SET PIRQ ERROR INTERRUPT VECTOR
6455  026662  063737  001004  000240          ADD     @#FACTOR,@#PIRVEC       ;ADD RELOCATION FACTOR
6456  026670  110337  177776                  MOVB    R3,@#PSW                ;SET CP PRIORITY LEVEL
6457  026674  050214                          BIS     R2,(R4)                 ;MAKE REQUEST AT LEVEL = TO CP LEVEL
6458  026676  100431                          BMI     5$                      ;BRANCH WHEN DONE
6459  026700  062737  000002  000240          ADD     #3$-2$,@#PIRVEC         ;SET PIRQ INTERRUPT VECTOR TO 3$
6460  026706  006302                          ASL     R2
6461  026710  050214                          BIS     R2,(R4)                 ;MAKE REQUEST AT LEVEL 1 HIGHER
6462  026712  000240                          NOP
6463  026714  104400                  2$:     HLT                             ;ERROR! EITHER AN INTERRUPT OCCURED
6464                                                                          ;WHEN RQST LEVEL = CP LEVEL (PIRVEC)=2$
6465                                                                          ;OR INTERRUPT FAILED (PIRVEC)=3$
6466  026716  022014                  3$:     CMP     (R0)+,(R4)              ;CHECK CONTENTS OF PIRQ REGISTER
6467  026720  001401                          BEQ     .+4
6468  026722  104400                          HLT                             ;ERROR! INCORRECT PIRQ CONTENTS
6469  026724  062703  000040                  ADD     #40,R3                  ;SET NEXT CP PRIORITY LEVEL
6470  026730  040214                          BIC     R2,(R4)                 ;LOWER LEVEL BY 1
6471  026732  012716  026654                  MOV     #1$,(SP)                ;ADJUST RETURN ADDRESS
6472  026736  063716  001004                  ADD     @#FACTOR,(SP)           ;TO RETURN TO 1$
6473  026742  000006                  30$:    RTT
6474
6475                                          ;TABLE OF CORRECT PIRQ REGISTER CONTENTS ON INTERRUPT
6476  026744  001042                  4$:     1042                            ;PIR1+PIA1
6477  026746  003104                          3104                            ;PIR2+PIR1+PIA2
6478  026750  007146                          7146                            ;PIR3+PIR2+PIR1+PIA3
6479  026752  017210                          17210                           ;PIR4+PIR3+PIR2+PIR1+PIA4
6480  026754  037252                          37252                           ;PIR5+PIR4+PIR3+PIR2+PIR1+PIA5
6481  026756  077314                          77314                           ;PIR6+PIR5+PIR4+PIR3+PIR2+PIR1+PIA6
6482  026760  177356                          177356                          ;PIR7+PIR6+PIR5+PIR4+PIR3+PIR2+PIR1+PIA7
6483
6484  026762  005014                  5$:     CLR     (R4)                    ;CLEAR PIRQ REGISTER
6485  026764  012737  000242  000240          MOV     #PIRVEC+2,@#PIRVEC      ;RESET PIRVEC TO HALT AT PIRVEC+2
6486  026772  005037  000242                  CLR     @#PIRVEC+2
6487  026776  105037  177776                  CLRB    @#PSW
6488  027002  012737  000006  000016          MOV     #6,@#TBITVEC+2          ;RESTORE 'T' BIT TRAP TO RETURN
6489  027010  012737  000016  000014          MOV     #TBITVEC+2,@#TBITVEC    ;VIA RTT IN TBITVEC+2
6490  027016  104000                          SCOPE
6491
6492                                          ;CHECK MFPI/MTPI INSTRUCTIONS
6493  027020  032737  140000  177776  MPI:    BIT     #UM,@#PSW               ;KERNEL MODE?
6494  027026  001537                          BEQ     ENDCP                   ;YES EXIT TEST
6495  027030  010746                          MOV     PC,-(SP)
6496  027032  062716  000144                  ADD     #5$-.,(SP)
6497  027036  012637  000250                  MOV     (SP)+,@#MMVEC           ;SET MEM MGMT ABORT VECTOR
6498  027042  005046                          CLR     -(SP)                   ;CLEAR CHECK WORD
6499  027044  010603                          MOV     SP,R3
6500  027046  010346                          MOV     R3,-(SP)                ;PUT ADDRESS OF CHECK WORD ON THE STACK
6501  027050  105737  000770                  TSTB    @#MMON                  ;CHECK IF MEM MGMT IS ENABLED
6502  027054  001423                          BEQ     1$                      ;BRANCH IF OFF
6503  027056  013737  177640  177654          MOV     @#UIPAR0,@#UIPAR6       ;SET UP USER PAGE ADDR. REG.
6504  027064  012737  006006  177614          MOV     #6006,@#UIPDR6          ;SET USER PAGE DESC REG R/W UP 6 PAGES
6505  027072  122737  000004  000766          CMPB    #4,@#OPT.CP             ;BRANCH IF 11/40
6506  027100  001406                          BEQ     10$
```

```
 6507  027102  013737  172240  172254          MOV     @#SIPAR0,@#SIPAR6
 6508  027110  012737  006006  172214          MOV     #6006,@#SIPDR6  ;SET SUPER PAGE DESC. REG.
 6509  027116  062706  140000          10$:    ADD     #140000,SP      ;SET CURRENT MODE'S STACK POINTER
 6510  027122  000240                          NOP
 6511  027124  010746                  1$:     MOV     PC,-(SP)
 6512  027126  062716  000024                  ADD     #3$-.,(SP)
 6513  027132  012637  000030                  MOV     (SP)+,@#EMTVEC  ;SET EMT TRAP VECTOR
 6514  027136  104000                          EMT                     ;TRAP TO 3$ BELOW
 6515  027140  005266  000002                  INC     2(SP)           ;INCREMENT CHECK WORD
 6516  027144  001417                          BEQ     6$
 6517  027146  104400                  4$:     HLT                     ;ERROR! MFPI,MTPI FAILURE-FOR BETTER
 6518  027150  000415                          BR      6$              ;ISOLATION SUGGEST RUNNING MFPI DIAG. DCKTD/E
 6519  027152  000240                  3$:     NOP                     ;PSW=KERNEL MODE,PREV USER OR SUPER MODE
 6520  027154  006506                          MFPI    SP              ;GET PREV. MODES STACK POINTER
 6521  027156  006536                          MFPI    @(SP)+          ;GET DATA (AN ADDRESS) ON PREV MODE'S STACK
 6522  027160  006576  000000                  MFPI    @(SP)           ;GET DATA (=0) FROM PREV MODES ADDRESS
 6523  027164  000240                          NOP                     ;SPACE AND PUSH ONTO KERNEL STACK
 6524  027166  001367                          BNE     4$              ;ERROR IF BRANCH TAKEN! SHOULD HAVE A ZERO ON THE STACK
 6525  027170  005116                          COM     (SP)            ;COMPLEMENT OPERAND
 6526  027172  006636                          MTPI    @(SP)+          ;POP OPERAND OFF KERNEL STACK AND MOVE
 6527                                                                  ;IT TO PREV MODE'S SPACE
 6528  027174  000002                          RTI                     ;RETURN TO INST FOLLOWING EMT ABOVE
 6529  027176  104400                  5$:     HLT                     ;ERROR! MEMORY MANG. ABORT
 6530  027200  105037  177776                  CLRB    @#PSW           ;SET PRIORITY LEVEL BACK TO 0
 6531  027204  012737  005510  000250  6$:     MOV     #KTABRT,@#MMVEC ;RESTORE VECTOR
 6532  027212  012737  001014  000030          MOV     #SCOPEA,@#EMTVEC
 6533  027220  012706  000500                  MOV     #STKPTR,SP      ;RESTORE STACK POINTER
 6534  027224  104000                          SCOPE
 6535
 6536                                  ;CHECK THAT HALT INSTRUCTION TRAPS TO 4 (11/45),10 (11/40) IN USER/SUPER MODE
 6537  027226  010746                  HALT1:  MOV     PC,-(SP)        ;GET CURRENT PC
 6538  027230  062716  000022                  ADD     #2$-.,(SP)
 6539  027234  011637  000004                  MOV     (SP),@#ERRVEC   ;SET ERROR TRAP VECTOR TO 2$ BELOW
 6540  027240  012637  000010                  MOV     (SP)+,@#RESVEC  ;LOAD RESERVED INST TRAP VECTOR (11/40)
 6541  027244  000000                          HALT                    ;SHOULD TRAP TO 4 IN USER/SUPER MODE
 6542  027246  104400                  1$:     HLT                     ;ERROR! HALT ABOVE FAILED IN USER/SUPER MODE
 6543  027250  000404                          BR      3$
 6544  027252  010716                  2$:     MOV     PC,(SP)         ;REPLACE RETURN PC WITH
 6545  027254  062716  000006                  ADD     #3$-.,(SP)      ;ADDRESS OF 3$ BELOW
 6546  027260  000002                          RTI                     ;RETURN (TO 3$)
 6547
 6548  027262  012737  005540  000004  3$:     MOV     #ERPRT,@#ERRVEC ;RESTORE ERROR TRAP VECTOR
 6549  027270  012737  005530  000010          MOV     #RESET,@#RESVEC
 6550  027276  104000                          SCOPE
 6551
 6552                                  ;TEST THAT RESET IS A 'NOP' IN USER/SUPER MODE
 6553  027300  000277                  RESET1: SCC
 6554  027302  013700  177776                  MOV     @#PSW,R0        ;GET CURRENT PSW
 6555  027306  000277                          SCC
 6556  027310  000005                          RESET
 6557  027312  023700  177776                  CMP     @#PSW,R0        ;CHECK THAT PSW UNCHANGED BY RESET ABOVE
 6558  027316  001401                          BEQ     .+4
 6559  027320  104400                          HLT                     ;ERROR! RESET CLEARED MODE BITS IN PSW
 6560  027322  010037  177776                  MOV     R0,@#PSW        ;RESTORE PSW (FOR ERROR)
 6561  027326  104000                  ENDCP:  SCOPE
 6562
```

```
6563  027330  010702                          MOV     PC,R2
6564  027332  062702  000012                  ADD     #12,R2
6565  027336  012707  001152                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
6566  027342  000000                  REL66:  .WORD   0
6567                                  ;6666666666666 LAST ADDRESS OF CODE TO BE RELOCATED 66666666666
6568
6569
6570                                          .SBTTL  START OF SECTION 7
6571                                  ;7777777777777 FIRST ADDRESS TO BE RELOCATED 777777777
6572  027344  010700                  REL7:   MOV     PC,R0           ;GET PC
6573  027346  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL7
6574  027350  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
6575  027354  012737  000007  005442          MOV     #7,@#SECT       ;SET SECTION #
6576  027362  004737  005472                  JSR     PC,@#LDDISP     ;LOAD DISPLAY GEG
6577  027366  013767  005486  001070          MOV     @#DISPLY,REL77
6578  027374  010700                          MOV     PC,R0           ;GET CURRENT PC
6579  027376  162700  027376                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
6580  027402  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
6581  027406  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR
6582
6583                                  ;TEST STACK LIMIT REGISTER
6584                                  ;THIS TEST SHIFTS A '1' BIT THROUGH ALL BIT POSITION
6585  027410  032737  004000  000766  STKLIM: BIT     #KJOPT,@#OPT.CP         ;CHECK IF OP.  N IS AVAILABLE
6586  027416  001512                          BEQ     10$             ;EXIT IF NOT .  ILABLE
6587  027420  012702  177774                  MOV     #SLR,R2         ;GET ADDRESS OF STACK LIM REG
6588  027424  005022                          CLR     (R2)+           ;CLEAR STACK LIMIT REG
6589  027426  032712  000020                  BIT     #T,(R2)         ;EXIT TEST IF 'T' BIT IS SET
6590  027432  001104                          BNE     10$
6591  027434  052712  000340                  BIS     #340,(R2)       ;SET PRIORITY LEVEL 7 TO PREVENT
6592                                                                  ;ANY INTERRUPTS FROM OCCURRING
6593  027440  012700  000400                  MOV     #400,R0         ;SET CHECK DATA
6594  027444  010042                  1$:     MOV     R0,-(R2)        ;MOVE TO STACK LIMIT REG
6595  027446  022200                          CMP     (R2)+,R0        ;AND CHECK RESULT
6596  027450  001401                          BEQ     2$
6597  027452  104400                          HLT                     ;ERROR! STACK LIMIT DID NOT
6598                                                                  ;LOAD CORRECTLY. CORRECT RESULT
6599                                                                  ;IS IN R0
6600  027454  006300                  2$:     ASL     R0              ;SHIFT '1' BIT LEFT
6601  027456  103372                          BCC     1$              ;LOOP UNTIL 1 BIT SHIFTS OUT
6602  027460  005042                          CLR     -(R2)           ;CLEAR STACK LIMIT REG
6603
6604                                  ;THIS TEST CHECKS THAT A PROPER 'RED' ZONE VIOLATION OCCURS, NOTE THAT
6605                                  ;NO 'RED ZONE' VIOLATION WILL OCCUR IF IN USER/SUPER MODES.
6606                                  ;A RED ZONE VIOLATION PUSHES THE CURRENT PSW,PC ON A STACK AT 2 AND 0
6607                                  ;AND TAKES THE NEXT INSTRUCTION FROM THE PC IN LOCATION4. THE INST-
6608                                  ;RUCTION CAUSING THE RED ZONE VIOLATION IS 'ABORTED'.
6609  027462  010746                          MOV     PC,-(SP)        ;GET CURRENT PC
6610  027464  062716  000054                  ADD     #4$-.,(SP)      ;FORM ADDRESS OF 4$ BELOW
6611  027470  012637  000004                  MOV     (SP)+,@#ERRVEC  ;SET ERROR TRAP VECTOR TO 4$ BELOW
6612  027474  013737  177776  000006          MOV     @#PSW,@#ERRVEC+2 ;RETAIN CURRENT STATUS ON TRAP
6613  027502  010712                          MOV     PC,(R2)         ;SET STACK LIMIT TO CURRENT PC
6614                                                                  ;+400
6615  027504  011206                          MOV     (R2),SP         ;AND STACK PTR = STACK LIMIT REG
6616  027506  010603                          MOV     SP,R3           ;SAVE STACK PTR
6617  027510  016304  000336                  MOV     336(R3),R4      ;SAVE MEMORY LOC CONTENTS
6618                                                                  ;AT 'RED ZONE' BOUNDARY
```

```
6619  027514  032737  140000  177776          BIT     #UM,@#PSW               ;BRANCH IF IN KERNEL MODE
6620  027522  001403                          BEQ     20$
6621  027524  010466  000336                  MOV     R4,336(SP)              ;SHOULD NOT CAUSE TRAP
6622  027530  000430                          BR      100$
6623
6624  027532  005066  000336          20$:    CLR     336(SP)                 ;SHOULD CAUSE 'RED ZONE' TRAP
6625  027536  104400                  3$:     HLT                             ;ERROR! FAILED TO TRAP
6626
6627  027540  032737  140000  000002  4$:     BIT     #UM,@#2                 ;CHECK IF TRAPPED WHEN IN USER
6628                                                                          ;/SUPER MODES (2 CONTAINS OLD PSW)
6629  027546  001013                          BNE     99$                     ;GO TO ERROR CALL
6630  027550  010600                          MOV     SP,R0                   ;STACK PTR SHOULD = 0
6631  027552  001011                          BNE     99$                     ;GO TO ERROR CALL IF NOT 0
6632  027554  026304  000336                  CMP     336(R3),R4              ;CHECK THAT INST WAS ABORTED
6633  027560  001006                          BNE     99$                     ;GO REPORT ERRPR
6634  027562  005012                  5$:     CLR     (R2)                    ;CLEAR STACK LIMIT REG
6635  027564  010705                          MOV     PC,R5                   ;GET CURRENT PC
6636  027566  062705  177750                  ADD     #3$-.,R5                ;FORM ADDRESS OF  3$ ABOVE
6637  027572  020516                          CMP     R5,(SP)                 ;CHECK THAT RETURN PC IS ON
6638                                                                          ;THE STACK (AT 0)
6639  027574  001406                          BEQ     100$                    ;EXIT TEST
6640
6641                                  ;ERROR
6642  027576  005012                  99$:    CLR     (R2)                    ;CLEAR STACK LIMIT REG
6643  027600  010463  000336                  MOV     R4,336(R3)              ;RESTORE MEM LOCATION
6644  027604  012706  000500                  MOV     #STKPTR,SP              ;SET STACK PTR
6645  027610  104400                          HLT                             ;ERROR!
6646  027612  010463  000336          100$:   MOV     R4,336(R3)              ;RESTORE MEM LOCATION
6647  027616  005022                          CLR     (R2)+                   ;CLEAR STACK LIM REG
6648  027620  012706  000500                  MOV     #STKPTR,SP              ;SET STACK PTR
6649  027624  042712  000340                  BIC     #340,(R2)               ;SET PRIORITY LEVEL BACK TO 0
6650  027630  012737  005540  000004          MOV     #ERPRT,@#ERRVEC         ;RESTORE ERROR TRAP VECTOR
6651  027636  012737  000002  000006          MOV     #RTI,@#ERRVEC+2
6652  027644  104000                  101$:   SCOPE
6653
6654                                  ;MEMORY MANAGEMENT REGISTER TESTS
6655                                  ;PDR TEST - THIS TEST WRITES 64. RANDOM #'S INTO EACH PDR REGISTER
6656                                  ;NOTE:  IF MEM MGMT IS ENABLED ONLY PDR/PAR PAIRS 2-6 ARE TESTED.
6657
6658  027646  005737  000766          KTPDR:  TST     @#OPT.CP                ;EXIT TEST IF NO KT OPTION
6659  027652  100151                          BPL     KTABT
6660  027654  012702  030142                  MOV     #PDRTBL,R2              ;SET TABLE ADDRESS OF PDR'S
6661  027660  012705  100360                  MOV     #100360,R5              ;SET BIT MASK (11/45)
6662  027664  005046                          CLR     -(SP)                   ;RESERVE LOCATION ON STACK
6663  027666  122737  000004  000766          CMPB    #4,@#OPT.CP             ;BRANCH IF 11/45
6664  027674  001005                          BNE     1$
6665  027676  005062  000004                  CLR     4(R2)                   ;TERMINATE TABLE AT SIPDR0
6666  027702  005062  000022                  CLR     22(R2)                  ;AND SIPAR0 (FOR FOLLOWING TEST)
6667  027706  005205                          INC     R5                      ;SET BIT MASK (11/40)
6668  027710  012200                  1$:     MOV     (R2)+,R0                ;GET PDR ADDRESS
6669  027712  001435                          BEQ     100$                    ;EXIT ON '0' TERMINATOR
6670  027714  012716  000010          2$:     MOV     #8.,(SP)                ;SET LOOP COUNT (FOR 8 REGS)
6671  027720  105737  000770                  TSTB    @#MMON                  ;BRANCH IF MEM MGMT ENABLED
6672  027724  001404                          BEQ     3$
6673  027726  062700  000004                  ADD     #4,R0                   ;SET R0 TO PDR2
6674  027732  012716  000004                  MOV     #4,(SP)                 ;AND LIMIT TO TEST 4 PDRS
```

```
 6675  027736  012703  000100          3$:     MOV     #64.,R3         ;SET DATA COUNT
 6676  027742  005004                  4$:     CLR     R4              ;INITIALIZE DATA TO BE WRITTEN
 6677  027744  040504                          BIC     R5,R4           ;CLEAR NON-SETTABLE BITS
 6678  027746  010410                          MOV     R4,(R0)         ;WRITE INTO PDR
 6679  027750  021004                          CMP     (R0),R4         ;AND CHECK DATA READ BACK
 6680  027752  001013                          BNE     99$             ;GO TO ERROR CALL
 6681  027754  005104                          COM     R4              ;COMPLEMENT DATA
 6682  027756  040504                          BIC     R5,R4           ;CLEAR NON-SETTABLE BITS
 6683  027760  010410                          MOV     R4,(R0)         ;WRITE COMPLEMENT DATA INTO PDR
 6684  027762  021004                          CMP     (R0),R4         ;AND CHECK
 6685  027764  001006                          BNE     99$             ;GO TO ERROR CALL
 6686  027766  060104                          ADD     R1,R4           ;STEP DATA
 6687  027770  077313                          SOB     R3,4$
 6688  027772  005020                  5$:     CLR     (R0)+           ;STEP TO NEXT REGISTER
 6689  027774  005316                          DEC     (SP)            ;DECREMENT REGISTER COUNT
 6690  027776  001357                          BNE     3$
 6691  030000  0  0743                        BR      1$              ;GET NEXT SET OF 8 REGISTERS
 6692
 6693  030002  10 4400                  99$:    HLT                     ;ERROR! INCORRECT DATA READ
 6694                                                                  ;BACK FROM PDR. ADDRESS OF
 6695                                                                  ;PDR IS IN R0, DATA IS IN R4
 6696  030004  000772                          BR      5$              ;STEP TO NEXT REGISTER
 6697  030006  005726                  100$:   TST     (SP)+           ;POP STACK
 6698  030010  104000                          SCOPE
 6699
 6700                                  ;PAR TEST - THIS TEST WRITES 64. COMPLEMENTING RANDOM #'S INTO EACH PAR.
 6701  030012  012702  030160          KTPAR:  MOV     #PARTBL,R2      ;GET TABLE ADDRESS OF PAR'S
 6702  030016  012705  170000                  MOV     #170000,R5      ;SET BIT MASK
 6703  030022  005046                          CLR     -(SP)           ;RESERVE LOCATION ON STACK
 6704  030024  122737  000010  000766        CMPB    #10,@#OPT.CP
 6705  030032  001001                          BNE     1$
 6706  030034  005005                          CLR     R5
 6707  030036  012200                  1$:     MOV     (R2)+,R0        ;GET PAR ADDRESS
 6708  030040  001435                          BEQ     100$            ;EXIT ON '0' TERMINATOR
 6709  030042  012716  000010          2$:     MOV     #8.,(SP)        ;SET LOOP COUNT (FOR 8 REGS.)
 6710  030046  105737  000770                  TSTB    @#MMON          ;BRANCH IF MEM MGMT ENABLED
 6711  030052  001404                          BEQ     3$
 6712  030054  062700  000004                  ADD     #4,R0           ;SET R0 TO PAR2
 6713  030060  012716  000004                  MOV     #4,(SP)         ;AND LIMIT TEST TO 4 PARS
 6714  030064  012703  000100          3$:     MOV     #64.,R3         ;SET DATA COUNT
 6715  030070  005004                          CLR     R4              ;INITIALIZE DATA
 6716  030072  040504                  4$:     BIC     R5,R4           ;CLEAR NON-SETTABLE BITS
 6717  030074  010410                          MOV     R4,(R0)         ;WRITE INTO PAR
 6718  030076  021004                          CMP     (R0),R4         ;AND CHECK
 6719  030100  001013                          BNE     99$             ;TAKE ERROR EXIT
 6720  030102  005104                          COM     R4              ;COMPLEMENT DATA
 6721  030104  040504                          BIC     R5,R4           ;CLEAR NON-SETTABLE BITS
 6722  030106  010410                          MOV     R4,(R0)         ;WRITE COMPLEMENT DATA
 6723  030110  021004                          CMP     (R0),R4         ;AND CHECK
 6724  030112  001006                          BNE     99$             ;TAKE ERROR EXIT
 6725  030114  060104                          ADD     R1,R4           ;STEP DATA
 6726  030116  077313                          SOB     R3,4$           ;LOOP UNTIL FINISHED
 6727
 6728  030120  005020                  5$:     CLR     (R0)+
 6729  030122  005316                          DEC     (SP)            ;DECREMENT REGISTER COUNT
 6730  030124  001357                          BNE     3$              ;BRANCH IF 8 REGS NOT DONE
```

```
6731  030126  000743                    BR      1$
6732
6733  030130  104400                    99$:    HLT                     ;ERROR! INCORRECT DATA READ BACK
6734                                                                      ;FROM PAR. ADDRESS OF PAR IS IN
6735                                                                      ;R0, DATA IS IN R4
6736  030132  000772                    BR      5$                      ;DO NEXT REGISTER
6737  030134  005726                    100$:   TST     (SP)+           ;POP STACK
6738  030136  104000                    SCOPE
6739  030140  000416                    BR      KTABT
6740                                    ;TABLES FOR PDR & PAR TESTS ABOVE
6741  030142  172300                    PDRTBL: .WORD   KIPDR0
6742  030144  177600                            .WORD   UIPDR0
6743  030146  172200                            .WORD   SIPDR0          ;CHANGED TO '0' IF 11/40
6744  030150  172320                            .WORD   KDPDR0
6745  030152  177620                            .WORD   UDPDR0
6746  030154  172220                            .WORD   SDPDR0
6747  030156  000000                            .WORD   0               ;TERMINATOR
6748
6749  030160  172340                    PARTBL: .WORD   KIPAR0
6750  030162  177640                            .WORD   UIPAR0
6751  030164  172240                            .WORD   SIPAR0          ;CHANGED TO '0' IF 11/40
6752  030166  172360                            .WORD   KDPAR0
6753  030170  177660                            .WORD   UDPAR0
6754  030172  172260                            .WORD   SDPAR0
6755  030174  000000                            .WORD   0               ;TERMINATOR
6756
6757                                    ;THIS TEST CHECKS KT ABORT LOGIC. TEST CREATES AN ABORT CONDITION
6758                                    ;AND INSURES THAT ABORT IS TAKEN PROPERLY. NOTE: TEST IS EXECUTED ONLY
6759                                    ;IF TEST IS ENTERED WITH MEM MGMT ENABLED.
6760  030176  105737  000770            KTABT:  TSTB    @#MMON          ;BRANCH IF MEM MGMT NOT
6761  030202  001523                            BEQ     KTEX            ;ENABLED
6762  030204  005037  172350                    CLR     @#KIPAR4        ;SET UP MEM MGMT REGISTERS
6763  030210  005037  172310                    CLR     @#KIPDR4        ;TO ABORT IF A MEMORY
6764  030214  005037  177650                    CLR     @#UIPAR4        ;REFERENCE IS MADE TO
6765  030220  005037  177610                    CLR     @#UIPDR4        ;ADDRESSES (VIRTUAL) BETWEEN
6766  030224  122737  000004  000766           CMPB    #4,@#OPT.CP     ;100000-117776 IN ALL MODES
6767  030232  001404                            BEQ     1$
6768  030234  005037  172250                    CLR     @#SIPAR4
6769  030240  005037  172210                    CLR     @#SIPDR4
6770  030244  013746  000250            1$:     MOV     @#MMVEC,-(SP)   ;SAVE MEM MGMT VECTOR
6771  030250  013746  000252                    MOV     @#MMVEC+2,-(SP) ;AND PRIORITY
6772  030254  010746                            MOV     PC,-(SP)        ;SET MEM MGMT
6773  030256  062716  000040                    ADD     #4$-.,(SP)      ;VECTOR TO 4$ BELOW
6774  030262  012637  000250                    MOV     (SP)+,@#MMVEC
6775  030266  013737  177776  000252           MOV     @#PSW,@#MMVEC+2
6776  030274  005000                            CLR     R0              ;CLEAR ABORT INDICATOR
6777  030276  010702                            MOV     PC,R2           ;SET R2 AND R3 NOTE:
6778  030300  012703  100000                    MOV     #100000,R3      ;THE REF VIA R3 CAUSES THE
6779  030304  014223                    2$:     MOV     -(R2),(R3)+     ;ABORT
6780  030306  005700                    3$:     TST     R0              ;BRANCH IF THE ABORT OCCURRED
6781  030310  001001                            BNE     .+4
6782  030312  104400                            HLT                     ;REPORT ERROR
6783  030314  000451                            BR      100$
6784                                    ;ABORT HERE
6785  030316  013700  177776            4$:     MOV     @#PSW,R0        ;SR0 SHOULD CONTAIN
6786  030322  000300                            SWAB    R0              ;CAUSE FOR ABORT AND
```

```
 6787  030324  006200                          ASR     R0              ;ALSO WHICH SEGMENT
 6788  030326  042700  177637                  BIC     #177637,R0      ;WAS IN USE WHEN ABORT
 6789  030332  062700  100011                  ADD     #100011,R0      ;OCCURRED.
 6790  030336  020037  177572                  CMP     R0,@#SR0
 6791  030342  001031                          BNE     99$
 6792  030344  012700  030304                  MOV     #2$,R0          ;GET ADDRESS OF INST THAT ABORTED
 6793  030350  020037  177576                  CMP     R0,@#SR2        ;THAT ABORTED
 6794  030354  001024                          BNE     99$
 6795  030356  122737  000004  000766          CMPB    #4,@#OPT.CP
 6796  030364  001414                          BEQ     5$
 6797  030366  012700  000362                  MOV     #362,R0
 6798  030372  120037  177574                  CMPB    R0,@#SR1        ;SR1 (11/45) CONTAINS REGISTER
 6799  030376  001013                          BNE     99$             ;MODIFICATIONS MADE
 6800  030400  012700  000023                  MOV     #23,R0
 6801  030404  120037  177575                  CMPB    R0,@#SR1+1
 6802  030410  001006                          BNE     99$
 6803  030412  012700  030304                  MOV     #2$,R0
 6804  030416  005720                  5$:     TST     (R0)+           ;R0=ADDRESS OF INST FOLLOWING ABORT
 6805  030420  020016                          CMP     R0,(SP)         ;(3$)
 6806  030422  001001                          BNE     99$
 6807  030424  000002                          RTI                     ;RETURN
 6808                                  ;ENTER HERE ON ERROR
 6809  030426  104400                  99$:    HLT                     ;REPORT ERROR
 6810  030430  010716                          MOV     PC,(SP)
 6811  030432  062716  177654                  ADD     #3$-.,(SP)
 6812  030436  000002                          RTI                     ;RETURN
 6813  030440  012637  000252          100$:   MOV     (SP)+,@#MMVEC+2 ;RESTORE ABORT VECTOR
 6814  030444  012637  000250                  MOV     (SP)+,@#MMVEC   ;& PRIORITY.
 6815  030450  104000                          SCOPE
 6816
 6817  030452                          KTEX:
 6818
 6819  030452  010702                          MOV     PC,R2
 6820  030454  062702  000012                  ADD     #12,R2
 6821  030460  012707  001152                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
 6822  030464  000000                  REL77:  .WORD   0
                                  ;7777777777777 LAST ADDRESS OF CODE TO BE RELOCATED 77777777777
 6823
 6824
 6825                                          .SBTTL  TELETYPE & CLOCK TESTS
 6826                                  ;CHECK TTY INTERRUPT.
 6827  030466  005037  001004          TTYCHK: CLR     @#FACTOR
 6828  030472  010701                          MOV     PC,R1
 6829  030474  032737  000400  000766          BIT     #TTOPT,@#OPT.CP ;BRANCH IT CONSOLE TTY AVAIL
 6830  030502  001002                          BNE     1$
 6831  030504  000167  000572                  JMP     CLKSET          ;JUMP IF NOT AVAILABLE
 6832  030510  032737  000100  177564  1$:     BIT     #100,@#TPS      ;CHECK IF TTY IS READY
 6833  030516  001374                          BNE     .-6
 6834  030520  012737  030574  000064          MOV     #3$,@#TPVEC     ;SET TTY INTERRUPT VECTOR
 6835  030526  012737  000200  000066          MOV     #200,@#TPVEC+2  ;PRIORITY LEVEL 4 ON INTERRUPT
 6836  030534  012767  030662  000114          MOV     #NULLS,MSG      ;ADDRESS OF MESSAGE TO BE TYPPED
 6837  030542  117737  000110  177566          MOVB    @MSG,@#TPB      ;TYPE FIRST CHARACTER OF MESSAGE
 6838  030550  105737  177564                  TSTB    @#TPS
 6839  030554  100375                          BPL     .-4
 6840  030556  006237  177564                  ASR     @#TPS           ;SET IE BIT IN TTY CSR REG
 6841  030562  000001                          WAIT                    ;WAIT FOR FIRST INTERRUPT
 6842  030564  000440                          BR      KW11
```

```
 6843  030566  006337  177564          2$:     ASL     @#TPS           ;CLEAR IE BIT
 6844  030572  000002                          RTI
 6845
 6846  030574  122777  000012  000054  3$:     CMPB    #12,@MSG        ;CHECK IF CHARACTER IS <LF>
 6847  030602  001020                          BNE     4$
 6848  030604  006337  177564                  ASL     @#TPS           ;CLEAR IE BIT
 6849  030610  052737  000340  177776          BIS     #PRTY7,@#PSW    ;SET PRIORITY LEVEL 7
 6850  030616  013746  177776                  MOV     @#PSW,-(SP)     ;PUSH PSW ONTO STACK
 6851  030622  004767  152110                  JSR     PC,.TYPE
 6852  030626  000752                          CRLF
 6853  030630  052737  000100  177564          BIS     #100,@#TPS      ;SET IE BIT
 6854  030636  005267  000014                  INC     MSG             ;STEP POINTER
 6855  030642  000002                          RTI
 6856  030644  117737  000006  177566  4$:     MOVB    @MSG,@#TPB      ;TYPE CHARACTER
 6857  030652  001745                          BEQ     2$              ;BRANCH IF TERMINATOR
 6858  030654  005227                  5$:     INC     (PC)+           ;SET MSG TO NEXT CHAR ADDRESS
 6859  030656  000000                  MSG:    .WORD   0               ;CONTAINS ADDRESS OF CHAR TO BE TYPED
 6860  030660  000002                          RTI                     ;RETURN
 6861  030662  020015  000015          NULLS:  .ASCIZ  <15><40><15>    ;CAR RET,SPACE,CAR RET.
 6862                                          .EVEN
 6863
 6864                                  ;ROUTINE TO TURN ON KW11-P OR KW11-L LINE CLOCK IF AVAILABLE
 6865  030666  010701                  KW11:   MOV     PC,R1
 6866  030670  012737  031176  000100          MOV     #LKSRV,@#LKVEC          ;LOAD INTERRUPT VECTOR
 6867  030676  012737  031226  000104          MOV     #PLKSRV,@#PLKVEC        ;FOR KW11-L & KW11-P CLOCKS
 6868  030704  012737  000300  000102          MOV     #300,@#LKVEC+2          ;SET PRIORITY LEVEL 6 ON INT.
 6869  030712  012737  000300  000106          MOV     #300,@#PLKVEC+2
 6870  030720  032737  002000  000766          BIT     #PLKOPT,@#OPT.CP        ;BRANCH IF 'P' CLOCK NOT AVAIL
 6871  030726  001407                          BEQ     10$
 6872  030730  012737  000002  172542          MOV     #2,@#PLKCSB             ;LOAD COUNT SET BUFFER
 6873  030736  012737  000101  172540          MOV     #101,@#PLKCSR           ;SET IE,100KHZ AND GO BITS
 6874  030744  000415                          BR      1$
 6875  030746  032737  001000  000766  10$:    BIT     #LKOPT,@#OPT.CP         ;BRANCH IF 'L' CLOCK NOT AVAIL
 6876  030754  001560                          BEQ     ARBEX                   ;SKIP PRIORITY ARBITRATION TEST
 6877                                                                          ;BELOW IF NO KW11-L OR KW11-P
 6878  030756  012737  000100  177546          MOV     #100,@#LKS              ;SET IE BIT
 6879  030764  012767  177546  000104          MOV     #LKS,6$
 6880  030772  012767  000240  000174          MOV     #NOP,9$
 6881
 6882                                  ;ROUTINE TO CHECK PRIORITY ARBITRATION LOGIC
 6883                                  ;THE BELOW TEST WILL INHIBIT INTERRUPTS ON LEVEL 6 AND ABOVE (LOCKING
 6884                                  ;OUT THE LINE CLOCK) AND THEN SET UP THE TTY TO INTERRUPT. NEXT THE
 6885                                  ;PRIORITY LEVEL WILL BE SET TO 0 ALLOWING INTERRUPTS IN WHICH CASE
 6886                                  ;THE LINE CLOCK (AT LEVEL 6) SHOULD INTERRUPT BEFORE THE TTY (AT LEVEL 4).
 6887  031000  132737  000020  177776  1$:     BITB    #20,@#PSW
 6888  031006  001143                          BNE     ARBEX                   ;EXIT TEST IF 'T' BIT SET
 6889  031010  032737  000100  177570          BIT     #100,@#SWR              ;BRANCH IF USER HAS DESELECTED
 6890  031016  001137                          BNE     ARBEX                   ;CLOCKS
 6891  031020  032737  000100  177564  2$:     BIT     #100,@#TPS              ;WAIT FOR TTY TO BE NOT
 6892  031026  001374                          BNE     2$                      ;BUSY
 6893  031030  112737  000300  177776          MOVB    #300,@#PSW              ;SET PRIORITY LEVEL 6
 6894  031036  152737  000100  177564  3$:     BISB    #100,@#TPS              ;SET IE BIT
 6895  031044  100374                          BPL     3$                      ;AND WAIT FOR EADY
 6896  031046  013767  000064  000210          MOV     @#TPVEC,.TPVEC          ;SAVE TTY VECTOR
 6897  031054  012737  031140  000064          MOV     #7$,@#TPVEC             ;SET TTY VECTOR
 6898  031062  005027                          CLR     (PC)+                   ;CLEAR CHECK WORD
```

```
 6899  031064  000000                  4$:     .WORD   0
 6900  031066  000240                          NOP
 6901  031070  000240                          NOP
 6902  031072  000240                          NOP
 6903  031074  113700                  5$:     MOVB    @(PC)+,R0               ;GET CLOCK STATUS & BRANCH IF READY
 6904  031076  172540                  6$:     .WORD   PLKCSR                  ;CONTAINS ADDRESS OF L OR P CLOCK STATUS REG.
 6905  031100  100375                          BPL     5$
 6906  031102  000240                          NOP                             ;AT THIS TIME BOTH THE CLOCK
 6907                                                                          ;AND THE TTY ARE READY TO INT
 6908  031104  012737  031154  000100          MOV     #8$,@#LKVEC             ;SET CLOCK VECTOR
 6909  031112  013737  000100  000104          MOV     @#LKVEC,@#PLKVEC
 6910  031120  105037  177776                  CLRB    @#PSW                   ;SET PRIORITY LEVEL 0
 6911
 6912  031124  022767  000002  177732          CMP     #2,4$                   ;CHECK THAT THE CLOCK
 6913  031132  001455                          BEQ     ARBFIN                  ;& TTY INTERRUPTED IN
 6914  031134  104400                          HLT                             ;THE PROPER SEQUENCE
 6915  031136  000453                          BR      ARBFIN
 6916
 6917  031140  042737  000100  177564  7$:     BIC     #100,@#TPS              ;CLEAR IE BIT
 6918  031146  006367  177712                  ASL     4$                      ;SHIFT INDICATOR
 6919  031152  000002                          RTI                             ;RETURN
 6920
 6921  031154  005267  177704          8$:     INC     4$
 6922  031160  012737  031176  000100          MOV     #LKSRV,@#LKVEC          ;SET CLOCK VECTORS
 6923  031166  012737  031226  000104          MOV     #PLKSRV,@#PLKVEC
 6924  031174  000414                  9$:     BR      PLKSRV                  ;FINISH SERVICE (NOTE: CONTAINS NOP IF NO P CLOC
 6925
 6926
 6927  031176  005267  147572          LKSRV:  INC     LTICKS                  ;INCREMENT CLOCK TICK COUNT
 6928  031202  012737  000100  177546          MOV     #100,@#LKS              ;CLEAR READY
 6929  031210  032737  000100  177570          BIT     #100,@#SWR              ;BRANCH IF USER DESIRES TO
 6930  031216  001402                          BEQ     1$                      ;KEEP CLOCK ENABLED
 6931  031220  005037  177546                  CLR     @#LKS                   ;DISABLE CLOCK
 6932  031224  000002                  1$:     RTI                             ;RETURN
 6933
 6934                                          ;KW11-P INTERRUPT SERVICE
 6935  031226  005267  147544          PLKSRV: INC     PTICKS
 6936  031232  012737  000100  172542          MOV     #100,@#PLKCSB
 6937  031240  012737  000101  172540          MOV     #101,@#PLKCSR           ;RE-ENABLE KW11-P
 6938  031246  032737  000100  177570          BIT     #100,@#SWR
 6939  031254  001402                          BEQ     1$
 6940  031256  005037  172540                  CLR     @#PLKCSR
 6941  031262  000002                  .$:     RTI
 6942
 6943  031264  000000                  .TPVEC: .WORD   0
 6944  031266  013737  031264  000064  ARBFIN: MOV     @#.TPVEC,@#TPVEC        ;RESTORE TTY VECTOR
 6945  031274  042737  000100  177564          BIC     #100,@#TPS
 6946  031302  012737  031176  000100  CLKSET: MOV     #LKSRV,@#LKVEC          ;SET CLOCK VECTORS
 6947  031310  012737  031226  000104          MOV     #PLKSRV,@#PLKVEC
 6948  031316  104000                  ARBEX:  SCOPE
 6949
 6950                                          ;TURN ON KW11-L CLOCK IF BOTH ARE AVAILABLE
 6951  031320  032737  001000  000766          BIT     #LKOPT,@#OPT.CP         ;BRANCH IF NOT AVAIL
 6952  031326  001411                          BEQ     1$
 6953  031330  012737  031176  000100          MOV     #LKSRV,@#LKVEC          ;SET VECTOR
 6954  031336  012737  000300  000102          MOV     #300,@#LKVEC+2          ;AND PRIOITY LEVEL 6 ON INT.
```

```
  6955  031344  052737  000100  177546          BIS     #100,@#LKS              ;SET IE BIT
  6956  031352                                  1$:
```

```
6957  031352  004767  000244            END:    JSR     PC,STMM         ;GO START RELOCATING ABOVE 28K
6958  031356  012737  005540  000004    END1:   MOV     #ERPRT,@#ERRVEC
6959  031364  005037  177776            CLR     @#PSW           ;CLEAR MODE BITS IN PSW
6960  031370  004767  151302            JSR     PC,CLRTBIT      ;GO CLEAR 'T' BIT IF SET
6961  031374  012706  000600            MOV     #KPTR,SP        ;SET KERNEL STACK PTR
6962  031400  032737  000400  000766    BIT     #TTOPT,@#OPT.CP ;BRANCH IF NO CONSOLE TTY
6963  031406  001440                    BEQ     4$
6964  031410  032737  000100  177564    BIT     #100,@#TPS      ;CHECK IF OUTPUT DEVICE IS BUSY
6965  031416  001374                    BNE     .-6             ;IS AVAILABLE
6966  031420  105737  177570            TSTB    @#SWR           ;DELETE END OF PASS TYPE OUT IF SW7=0
6967  031424  100024                    BPL     1$              ;BRANCH IF SW7 IS DOWN
6968  031426  016702  147346            MOV     ICNT,R2         ;GET PASS COUNT
6969  031432  004767  151420            JSR     PC,CNVDAT               ;CONVERT TO ASCIZ STRING
6970  031436  012702  003262            MOV     #DIGITS+2,R2    ;GET ASCII VALUES
6971  031442  012703  004010            MOV     #PASSES,R3      ;AND MOVE THEM INTO MESSAGE
6972  031446  012704  004164            MOV     #PASSNO,R4
6973  031452  011223                    MOV     (R2),(R3)+
6974  031454  012224                    MOV     (R2)+,(R4)+
6975  031456  011223                    MOV     (R2),(R3)+
6976  031460  012224                    MOV     (R2)+,(R4)+
6977  031462  012737  004056  030656    MOV     #SUCCESS,@#MSG  ;PASS MESSAGE ADRS TO TELETYPE SERVICE
6978  031470  052737  000100  177564    BIS     #100,@#TPS      ;SET IE BIT
6979  031476  105737  177570      1$:   TSTB    @#SWR
6980  031502  100402                    BMI     4$
6981  031504  000004  033004            TYPE,ENDNS
6982  031510  005267  147264      4$:   INC     ICNT
6983  031514  116700  147246            MOVB    OPT.CP,R0       ;GET CP TYPE
6984  031520  026067  032676  1?        CMP     PASTAB(R0),ICNT ;CHECK IF END OF TEST
6985  031526  001002                    BNE     2$              ;BRANCH IF NOT AT END
6986  031530  000167  001006            JMP     DONE
6987  031534  016702  147240      2$:   MOV     ICNT,R2         ;GET PASS COUNT
6988  031540  006302                    ASL     R2
6989  031542  046002  032652            BIC     CPPASS(0),R2    ;LIMIT PASS COUNT TO 0-6
6990  031546  012737  000016  000014    MOV     #16,@#14        ;RESTORE VECTOR  FOR RTT TEST
6991  031554  005037  000016            CLR     @#16            ;CLEAR T BIT TRAP ADDRESS
6992  031560  113737  001151  001150    MOVB    @#ITCNT+1,@#ITCNT       ;RESET ITERATION COUNT
6993  031566  016216  032632            MOV     PSWTAB(2),(SP)  ;PUSH NEXT PASS PSW ON STACK
6994  031572  032716  000020            BIT     #20,(SP)        ;WILL 'T' BIT BE SET ON NEXT PASS?
6995  031576  001406                    BEQ     3$              ;BRANCH IF NOT
6996  031600  112737  000001  001150    MOVB    #1,@#ITCNT      ;SET ITERATION COUNT = 1 FOR 'T' BIT
6997  031606  016737  000006  000016    MOV     RTI1,@#16       ;SET 'T' BIT TRAP TO RETURN VIA 16
6998  031614  012746  006120      3$:   MOV     #START2,-(SP)   ;RESART PROGRAM AT START2
6999  031620  000006              RTI1: RTT                     ;RESTART PROGRAM AT START2 WITH NEW PSW
7000                                                          ;(FROM TABLE BELOW)
7001
```

```
                                                .SBTTL  STMM ROUTINE
                                        ;ROUTINE TO SET UP MEMORY MANAGEMENT TO RELOCATE PROGRAM CODE ABOVE 28K
031622  005737  000766          STMM:   TST     @#OPT.CP        ;CHECK FOR MEM MGMT OPTION
031626  100401                  1$:     BMI     2$              ;BRANCH IF AVAILABLE
031630  000207                          RTS     PC              ;RETURN

                                        ;CHECK IF PROGRAM IS TO BE RELOCATED ABOVE 28K.
                                        ;SW12,SW09=0,1 OR SW12,SW09=1,0 = NO RELOCATION
031632  032737  001000  177570  2$:     BIT     #SW09,@#SWR     ;BRANCH IF SW09 IS = 0
031640  001406                          BEQ     3$
031642  032737  010000  177570          BIT     #SW12,@#SWR     ;JUMP IF NO RELOCATION ABOVE 28K
031650  001010                          BNE     4$              ;I.E. SW12,SW09=1,0
031652  000167  000624                  JMP     ENDM            ;RETURN TO END1 VIA ENDM

031656  032737  010000  177570  3$:     BIT     #SW12,@#SWR     ;BRANCH IF SW12=0
031664  001402                          BEQ     4$
031666  000167  000610                  JMP     ENDM            ;RETURN TO END1 VIA ENDM

                                        ;THE PROGRAM IS GOING TO RELOCATE.
                                        ;RELOCATION WILL BE FROM PHYSICAL 0 - LAST ADDRESS (VIA KIPAR0/2) TO
                                        ;PHYSICAL ADDRESS FORMED USING KIPAR2/KIPAR3.
                                        ;RELOCATION WILL BE PERFORMED IN KERNEL MODE WITH PSW SET AT PRIORITY
                                        ;LEVEL 4 (TO PREVENT TTY INTERRUPT-WHICH CHANGES DATA IN PROGRAM)
                                        ;THE 'T' BIT IS CLEARED (IF SET). AFTER THE DATA HAS BEEN WRITTEN IT IS
                                        ;VERIFIED BEFORE EXECUTION.
031672  013727  177776          4$:     MOV     @#PSW,(PC)+     ;SAVE CURRENT PSW
031676  000000                  OLDPSW: .WORD   0
031700  012737  000200  177776          MOV     #PRTY4,@#PSW    ;SET LEVEL 4 & KERNEL MODE
031706  004767  150764                  JSR     PC,CLRTBIT      ;GO CLEAR 'T' BIT IF SET

                                        ;NOW SETUP MEMORY MANAGEMENT REGISTERS.
031712  010701                          MOV     PC,R1           ;SET SCOPE PTR
031714  012700  077406                  MOV     #77406,R0       ;SET CONSTANT=R/W UP 4K WORDS
031720  010037  172300                  MOV     R0,@#KIPDR0     ;SET KIPDR0,1,2,3,& 7 R/W UP 4K WORDS
031724  010037  172302                  MOV     R0,@#KIPDR1
031730  010037  172304                  MOV     R0,@#KIPDR2
031734  010037  172306                  MOV     R0,@#KIPDR3
031740  010037  172316                  MOV     R0,@#KIPDR7

031744  005037  172340                  CLR     @#KIPAR0        ;NOTE: THESE 2 INSTRUCIONS EFFECTIVELY
031750  012737  000200  172342          MOV     #200,@#KIPAR1   ;RELOCATE PROGRAM EXECUTION
031756  016737  000450  172344          MOV     NEXPAR,@#KIPAR2 ;SET UP KIPAR2 & KIPAR3
031764  013737  172344  172346          MOV     @#KIPAR2,@#KIPAR3
031772  062737  000200  172346          ADD     #200,@#KIPAR3
032000  012737  177600  172356          MOV     #177600,@#KIPAR7;AND OF COUSE THE I/O PAGE
032006  012737  000001  177572          MOV     #1 @#SR0        ;ENABLE MEM MGMT
032014  005046                          CLR     -(SP)
032016  032737  011000  177570          BIT     #SW12!SW09,@#SWR        ;BRANCH IF NO RELOCATION
032024  001006                          BNE     1$              ;ABOVE 128K

032026  122737  000010  000766          CMPB    #10,@#OPT.CP
032034  001002                          BNE     1$
032036  012716  000020                  MOV     #20,(SP)

                                        ;NOW SETUP USER MEM MGMT REGISTERS
032042  010037  177600          1$:     MOV     R0,@#UIPDR0     ;SET UP USER MEM MGMT REGS
```

```
7058  032046  010037  177602              MOV     R0,@#UIPDR1
7059  032052  010037  177616              MOV     R0,@#UIPDR7
7060  032056  016737  000350  177640      MOV     NEXPAR,@#UIPAR0
7061  032064  013737  177640  177642      MOV     @#UIPAR0,@#UIPAR1
7062  032072  062737  000200  177642      ADD     #200,@#UIPAR1
7063  032100  013737  172356  177656      MOV     @#KIPAR7,@#UIPAR7
7064
7065                                ;CHECK IF 11/40 & IF NOT SETUP SUPERVISOR REGISTERS
7066  032106  122737  000004  000766      CMPB    #4,@#OPT.CP     ;BRANCH IF AN 11/40
7067  032114  001424                      BEQ     3$
7068  032116  010037  172200              MOV     R0,@#SIPDR0     ;SET UP SUPERVISOR MEM MGMT REGS
7069  032122  010037  172202              MOV     R0,@#SIPDR1
7070  032126  010037  172216              MOV     R0,@#SIPDR7
7071  032132  016737  000274  172240      MOV     NEXPAR,@#SIPAR0
7072  032140  013737  172240  172242      MOV     @#SIPAR0,@#SIPAR1
7073  032146  062737  000200  172242      ADD     #200,@#SIPAR1
7074  032154  013737  172356  172256      MOV     @#KIPAR7,@#SIPAR7
7075  032162  011637  172516              MOV     (SP),@#SR3              ;SETUP SR3
7076
7077  032166  005726                3$:   TST     (SP)+                   ;POP STACK
7078  032170  000240                      NOP
7079  032172  110637  000770              MOVB    SP,@#MMON       ;SET MEM MGMT ON IND = ON
7080  032176  004767  153230              JSR     PC,LDDISP       ;LOAD DISPLAY REGISTER
7081  032202  013767  005436  000600      MOV     @#DISPLY,ENDTAG ;AND ALSO LAST WORD XFERED
7082  032210  012737  032500  000004      MOV     #ENDMEM,@#ERRVEC;SET TIME OUT TRAP VECTOR
7083
7084                                ;IF AN ERROR OCCURS ON RELOCATION THE PROGRAM WILL RETRY THE OPERATION
7085                                ;UNTIL THE ERROR NO LONGER OCCURS.
7086  032216  012702  040000        RETRY:  MOV     #40000,R2       ;SETUP GENERAL REGISTERS
7087  032222  005000                        CLR     R0              ;DATA WILL BE RELOCATED FROM
7088                                                                ;ADDRESS IN R0 TO ADDRESS IN R2
7089  032224  012704  033012              MOV     #ENDTAG+2,R4    ;GET # OF BYTES TO RELOCATE
7090  032230  010203                      MOV     R2,R3           ;GET 'TO' ADDRESS
7091  032232  060403                      ADD     R4,R3           ;FORM LAST 'TO' ADDRESS FOR RELOCATION
7092  032234  010013                      MOV     R0,(R3)         ;TRAP TO ENDMEM IF INSUFFICIENT MEMORY
7093  032236  012737  005540  000004      MOV     #ERPRT,@#ERRVEC ;RESTORE ERROR TRAP VECTOR
7094
7095                                ;RELOCATION MAY BE PERFORMED BY AN I/O DEVICE OR THE CP. WHICH IS IT?
7096  032244  105737  000771              TSTB    @#QV
7097  032250  001013                      BNE     11$
7098  032252  032737  000040  177570      BIT     #SW05,@#SWR     ;BRANCH IF ALL DEVICES DESIRED FOR
7099  032260  001007                      BNE     11$             ;RELOCATION ROUND ROBIN STYLE
7100  032262  032737  000010  177570      BIT     #SW03,@#SWR     ;BRANCH IF RELOCATION IS VIA CP
7101  032270  001431                      BEQ     1$
7102  032272  113737  177570  000757      MOVB    @#SWR,@#DEV     ;GET SELECTED DEVICE IN <SW02-SW00>
7103
7104                                ;IF AN I/O DEVICE IS SELECTED THE PROGRAM MUST FORM AN 18 PHYS ADDRESS.
7105  032300  005046                11$:   CLR     -(SP)           ;CLEAR WORKING LOCATION
7106  032302  013702  172344              MOV     @#KIPAR2,R2     ;FORM ADDRESS FOR READ DATA
7107  032306  006302                      ASL     R2              ;SHIFT KIAPR BITS TO FORM
7108  032310  006302                      ASL     R2              ;18 BIT PHYSICAL ADDRESS
7109  032312  006302                      ASL     R2              ;IN R2 AND TOP OF STACK
7110  032314  006302                      ASL     R2
7111  032316  006302                      ASL     R2
7112  032320  006116                      ROL     (SP)
7113  032322  006302                      ASL     R2
```

```
7114  032324  006116                          ROL     (SP)
7115  032326  006316                          ASL     (SP)            ;POSITION EA BITS AT
7116  032330  006316                          ASL     (SP)            ;BIT POSITION
7117  032332  006316                          ASL     (SP)            ;4 AND 5
7118  032334  006316                          ASL     (SP)
7119  032336  112637  000762                  MOVB    (SP)+,@#EABITS  ;AND SAVE IN EABITS
7120  032342  004737  001430                  JSR     PC,@#IODEV      ;GO RELOCATE DATA VIA I/O DEVICE
7121  032346  102005                          BVC     10$             ;BRANCH IF NO DEVICE ERROR
7122  032350  012702  040000                  MOV     #40000,R2       ;RESTORE 'TO' ADDRESS
7123  032354  012022                  1$:     MOV     (R0)+,(R2)+     ;RELOCATE CODE TO ADDRESS SPECIFIED
7124                                                                  ;IN R2 (KIPAR2/KIPAR3)
7125  032356  020302                          CMP     R3,R2           ;CHECK IF AT LAST ADDRESS
7126  032360  001375                          BNE     1$
7127  032362  010302                  10$:    MOV     R3,R2           ;GET VALUE OF LAST ADDRESS
7128  032364  012703  001000                  MOV     #1000,R3        ;DO NOT CHECK FIRST 1000 (8) LOCATIONS
7129  032370  024042                  2$:     CMP     -(R0),-(R2)     ;CHECK THAT DATA WAS RELOCATED PROPERLY
7130  032372  001403                          BEQ     3$
7131  032374  004737  002646                  JSR     PC,@#SAVVAL     ;GO SAVE APPROPRIATE VALUES
7132  032400  104400                          HLT                     ;ERROR! DATA NOT RELOCATED PROPERLY
7133                                                                  ;R0= SOURCE/R2=DEST ADDRESS
7134  032402  020003                  3$:     CMP     R0,R3           ;BRANCH IF NOT AT LAST ADDRESS
7135  032404  001371                          BNE     2$
7136  032406  162737  000010  000772          SUB     #10,@#DEVID     ;BRANCH IF ERROR ON RELOCATION
7137  032414  001700                          BEQ     RETRY
7138  032416  105237  000757                  INCB    @#DEV           ;STEP TO NEXT DEVICE
7139  032422  005037  000772                  CLR     @#DEVID         ;SET DEVICE IND = CP
7140  032426  062727  000040                  ADD     #40,(PC)+       ;SET VALUE FOR NEXT RELOCATION
7141  032432  000000                  NEXPAR: .WORD   0               ;CONTAINS NEXT VALUE FOR KIPAR2
7142  032434  013737  172344  172340          MOV     @#KIPAR2,@#KIPAR0       ;NOTE: THESE 2 INSTRUCTIONS RE-
7143  032442  013737  172346  172342          MOV     @#KIPAR3,@#KIPAR1       ;LOACATE PROGRAM EXECUTION
7144                                  ;*****************************************************************
7145                                  ;*****************************************************************
7146                                  ;PROGRAM IS NOW EXECUTING IN KERNEL MODE RELOCATED TO ADDRESS AS SPEC-
7147                                  ;IFIED IN KIPAR0. FOR EX. IF KIPAR0=1600 THEN PROGRAM EXECUTING AT
7148                                  ;ADDRESS 160000+(PC)
7149  032450  012706  000600                  MOV     #KPTR,SP        ;SET KERNEL STACK PTR
7150  032454  005037  177776                  CLR     @#PSW
7151  032460  016746  177212                  MOV     OLDPSW,-(SP)    ;RESTORE OLD PSW
7152  032464  012746  032472                  MOV     #1$,-(SP)
7153  032470  000002                          RTI
7154
7155  032472  000240                  1$:     NOP                     ;DON'T REPLACE WITH HALT IF USER/SUPER MODE
7156  032474  000137  006120                  JMP     @#START2        ;RESTART PROGRAM AT START2
7157
7158                                  ;WHEN RELOCATION ABOVE 28K IS COMPLETE PROGRAM TRAPS TO ENDMEM.
7159  032500  022626                  ENDMEM: CMP     (SP)+,(SP)+     ;POP STACK TWICE
7160  032502  005037  177572          ENDM:   CLR     @#SR0           ;DISABLE MEM MGMT
7161  032506  122737  000004  000766          CMPB    #4,@#OPT.CP     ;BRANCH IF 11/40
7162  032514  001402                          BEQ     1$
7163  032516  005037  172516                  CLR     @#SR3
7164  032522  000240                  1$:     NOP
7165
7166                                  ;*****************************************************************
7167                                  ;*****************************************************************
7168                                  ;AT THIS TIME A 'PASS' HAS BEEN COMPLETED.
7169                                  ;PROGRAM NOW EXECUTING IN KERNEL MODE AT PC AS SHOWN (NO RELOCATION)
```

```
7170  032524  012767  001600  177700          MOV     #1600,NEXPAR    ;RESET NEXT VALUE FOR PAR REGISTERS
7171  032532  105037  000770                  CLRB    @#MMON          ;SET MEM MGMT ON IND = OFF
7172  032536  000137  031356                  JMP     @#END1          ;RETURN TO INST FOLLOWING JSR PC,LDMM
7173
7174
7175                                          ;WHEN THE PROGRAM HAS COMPLETED THE REQUISITE # OF PASSES ENTER HERE.
7176  032542  032737  000100  177564  DONE:   BIT     #100,@#TPS      ;WAIT FOR TTY OUTPUT TO FINISH
7177  032550  001374                          BNE     DONE
7178  032552  105037  177566                  CLRB    @#TPB           ;TYPE NULL CHARACTER
7179  032556  105737  177564                  TSTB    @#TPS           ;WAIT UNTIL DONE
7180  032562  100375                          BPL     .-4
7181  032564  005000                          CLR     R0
7182  032566  162700  000001          1$:     SUB     #1,R0
7183  032572  001375                          BNE     1$
7184  032574  000005                          RESET
7185  032576  105737  177570                  TSTB    @#SWR           ;BRANCH IF NOT TYPECUT DESIRED
7186  032602  100002                          BPL     2$
7187  032604  000004  032741                  TYPE,ENDMSG
7188  032610  013702  000042          2$:     MOV     @#42,R2         ;CHECK DDP/ACT11 MONITOR HOOK
7189  032614  001404                          BEQ     DONE1
7190  032616  004712                  LOGICAL:JSR     PC,(R2)         ;GO TO DDP/ACT11 MONITOR VIA 42
7191  032620  000240                          NOP
7192  032622  000240                          NOP
7193  032624  000240                          NOP
7194  032626  000137  006050          DONE1:  JMP     @#START3        ;RESTART PROGRAM
7195
7196                                          ;THE BELOW TABLE REPRESENTS THE 'NEW' PSW SET BY THE PROGRAM ON
7197                                          ;SUCCESSIVE PASSES.
7198                                          ;NOTE THE BELOW TABLE MAY BE MODIFIED TO CAUSE THE PROGRAM TO RUN
7199                                          ;UNDER USER DEFINED PARAMETERS BY PATCHING IN THE DESIRED PASS PARAMETER
7200                                          ;FOR EXAMPLE TO CAUSE THE PROGRAM TO RUN WITHOUT SETTING THE 'T' BIT
7201                                          ;IN ALL PASSES PATCH OUT THE 'T' BIT IN THE TABLE.
7202  032632  000000                  PSWTAB: 000000                  ;ALL 11 FAMILY CP'S
7203  032634  000020                          000020
7204  032636  140000                          140000                  ;11/45, 11/40 ONLY
7205  032640  140020                          140020
7206  032642  144000                          144000                  ;11/45 ONLY
7207  032644  144020                          144020
7208  032646  044000                          044000
7209  032650  044020                          044020
7210
7211
7212                                          ;THE BELOW TABLE IS THE 'BIT MASK' USED TO DETERMINE THE INDEX VALUE
                                              ;NEEDED TO SET THE 'NEW' PSW.
7213  032652  177774                  CPPASS: 177774
7214  032654  177774                          177774
7215  032656  177770                          177770                  ;11/40
7216  032660  177760                          177760                  ;11/45
7217  032662  177760                          177760
7218
```

```
7219                                    ;THE BELOW TABLE REPRESENTS THOSE BITS IN THE CP WHICH CAN BE SET/CLEARED
7220                                    ;EXCLUDING THE REGISTER SET BIT IN THE 11/45.
7221  032664  000377                    PSWBIT: 000377
7222  032666  000377                            000377
7223  032670  000357                            000357                  ;11/40
7224  032672  170357                            170357                  ;11/45 (REGSET BIT IS CHECKED ELSEWHERE)
7225  032674  170357                            170357
7226
7227                                    ;THE BELOW TABLE CONTAINS THE # OF PASSES REQUIRED TO COMPLETE TEST
7228  032676  000002                    PASTAB: .WORD   2
7229  032700  000002                            .WORD   2
7230  032702  000004                            .WORD   4               ;11/40
7231  032704  000010                            .WORD   10              ;11/45
7232  032706  000010                            .WORD   10
7233
7234                                    ;MESSAGES
7235  032710  005015  047514  020127    MSG1:   .ASCIZ  <15><12>'LOW LIMIT?'
7236  032716  044514  044515  037524
7237  032724     000
7238  032725     110  043511  020110    MSG2:   .ASCIZ  'HIGH LIMIT?'
7239  032732  044514  044515  037524
7240  032740     000
7241  032741     015  020012  041504    ENDMSG: .ASCIZ  <15><12>' DCQKC DONE' <7><0>
7242  032746  045521  020103  047504
7243  032754  042516  000007     000
7244          032762                            .EVEN
7245  032762  005015                    NODEV:  .ASCII  <15><12>
7246  032764  054130  047040  052117    DEVNAM: .ASCIZ  'XX NOT ON BUS'<15><12>
7247  032772  047440  020116  052502
7248  033000  006523  000012
7249
7250
7251
7252  033004  025007     000            ENDNS:  .ASCIZ  <7><52>
7253          033010                            .EVEN
7254                                            .SBTTL  SUBROUTINE CHECKS
7255  033010  000000                    ENDTAG: .WORD   0
7256                                    ;NOTE: THE BELOW CODE GETS OVERLAID WHEN THE PROGRAM IS STARTED.
7257
7258  033012  005015  041504  045521    A. T.CP:.ASCII  <15><12>'DCQKC-G 11/40 & 11/45 INST EXER'
7259  033020  026503  020107  030461
7260  033026  032057  020060  020046
7261  033034  030461  032057  020065
7262  033042  047111  052123  042440
7263  033050  042530     122
7264  033053     015  047412  052120            .ASCIZ  <15><12>'OPT.CP='
7265  033060  041456  036520     000
7266          033066                            .EVEN
7267
7268  033066  005015  050117  027124    OPTDEV: .ASCIZ  <15><12>'OPT.DEV.='
7269  033074  042504  027126  000075
7270                                            .EVEN
7271
7272                                    ;THESE ROUTINES ARE USED TO CHECK THE TYPE,CONVERT,HLT,AND SCOPE ROUTINES.
7273                                    ;CHECK TYPE ROUTINE:
7274  033102  012706  000500            CHKTYP: MOV     #500,SP         ;SET STACK PTR
```

```
 7275  033106  000004  004056                  TYPE,SUCCESS
 7276  033112  000773                          BR      CHKTYP
 7277
 7278  033114  012706  000500  CHKCNV:         MOV     #500,SP         ;SET STACK PTR
 7279  033120  012702  123456                  MOV     #123456,R2      ;SET VALUE TO BE CONVERTED & TYPED
 7280  033124  004767  150140                  JSR     PC,TYPDAT       ;TYPE 123456
 7281  033130  000004  000752                  TYPE,CRLF
 7282  033134  000767                          BR      CHKCNV
 7283
 7284  033136  012706  000500  CHKHLT:         MOV     #500,SP         ;SET STACK PTR
 7285  033142  104400                          HLT
 7286  033144  000774                          BR      CHKHLT
 7287
 7288  033146  012706  000500  CHKSCP:         MOV     #500,SP             ;SET STACK PTR
 7289  033152  005000                          CLR     R0
 7290  033154  010701                          MOV     PC,R1               ;SET SCOPE PTR
 7291  033156  010037  177570                  MOV     R0,@#DISPLAY
 7292  033162  005200                          INC     R0
 7293  033164  104000                          SCOPE
 7294  033166  000767                          BR      CHKSCP
 7295
 7296
 7297          000001                          .END
```

```
AC0     =X000000          1074#
AC1     =X000001          1075#
AC2     =X000002          1076#
AC3     =X000003          1077#
AC4     =X000004          1078#
AC5     =X000005          1079#
ADC82     012344          3413    3415#
ADC85     013154          3642    3644#
ADC86     013642          3794    3795    3797#
ADC87     014510          4005    4006    4007    4009#
ADC0      010304          2824    2825    2826    2828#
ADC1      011160          3047    3048    3049    3051#
ADC2      012154          3345    3347#
ADC5      012762          3569    3570    3572#
ADC6      013452          3739    3740    3742#
ADC7      014404          3969    3970    3972#
ADD0      015176          4170    4171    4172    4174#
ADD1      015274          4206    4207    4209#
ADD1A     015520          4289    4290    4291    4293#
ADD1B     015536          4298    4299    4301#
ADD2      016124          4435    4436    4438#
ADD3      016672          4627    4629#
ADD6      017234          4724    4725    4727#
ADD7      017676          4836    4837    4838    4840#
ADRSIS    005210          2269    2318#
ADRTAB    002312          1701    1706    1716#
AOPT.C    033012          2548    7258#
ARBEX     031316          6876    6888    6890    6948#
ARBFIN    031266          6913    6915    6944#
ASCAN     005275          2309    2328#
ASHCL0    025670          6223#
ASHCR0    025746          6244#
ASHL0     025460          6169    6173#
ASHL1     026252          6346    6350#
ASHR0     025574          6200#
ASHR1     026340          6373#
ASLB1     011522          3179    3180    3182#
ASLB1A    011746          3266    3267    3269#
ASLB3     013144          3636    3637    3639#
ASLB4     012450          3450    3451    3452    3454#
ASLB6     013624          3786    3787    3788    3790#
ASLB7     014606          4037    4038    4040#
ASL0      010426          2868    2869    2870    2871    2873#
ASL1      011334          3110    3111    3112    3114#
ASL3      012676          3538    3539    3541#
ASL4      012246          3377    3378    3379    3381#
ASL6      013422          3727    3728    3730#
ASL7      014232          3916    3917    3919#
ASRB1     011616          3215    3217#
ASRB1A    011632          3221    3222    3224#
ASRB2     012414          3435    3436    3438#
ASRB2A    012432          3443    3444    3446#
ASRB5     013104          3617    3618    3620#
ASRB6     013742          3826    3827    3829#
ASRB7     014624          4044    4045    4047#
ASR0      010454          2882    2883    2884    2886#
```

```
ASR1    011222          3067    3068    3069    3071#
ASR2    012170          3351    3352    3354#
ASR3    012662          3532    3534#
ASR6    013304          3689    3690    3692#
ASR7    014266          3930    3931    3933#
A.DATA  004402          2149    2156    2169#
BORDR   004410          2152    2170#
BODAT   005354          2294    2337#
BELL    004054          2072    2110#
BICB1   015712          4359    4360    4362#
BICB1A  015734          4370    4373#
BICO    015110          4141    4142    4143    4145#
BIC1    015416          4252    4253    4255#
BIC2    016214          4464    4465    4466    4468#
BIC3    016704          4632    4634#
BIC7    020500          5006#   5008
BINB    016434          4528    4530    4532    4535    4538    4540    4543    4546#
BINB7   020246          4941    4949#
BIN1    016072          4402    4405    4408    4411    4414    4417    4420    4425#
BISB1   015700          4354    4356#
BISO    015066          4132    4133    4135#
BISOA   015144          4159    4161#
BIS1    015404          4246    4247    4249#
BIS2    016152          4448    4450#
BIS2A   016254          4482    4483    4485#
BIS7    020440          4996#   4998
BITB1   015670          4348    4349    4351#
BITB2   016522          4570    4571    4573#
BITB3   017054          4684    4687#
BITB6   017440          4771#
BIT1    015332          4223    4224    4226#
BIT13 = 020000          1312#
BIT14 = 040000          1311#
BIT15 = 100000          1310#
BIT2    016240          4475    4476    4478#
BIT6  = 000100          1314#
BIT8  = 000400          1313#
BPTVEC= 000014          1111#   5423    5426*   5427*   5499*   5500*
BUSADR  001372          1521#   1634*   1659*
C     = 000001          1087#
CBIT    023466          5672#
CCO     010120          2748    2749    2750    2751    2752    2753    2754    2756#
CC1     010134          2760    2761    2762    2764#
CC2     010150          2768    2769    2770    2772#
CC3     010162          2776    2777    2779#
CC4     010176          2783    2784    2785    2787#
CHKCNV  033114          7278#   7282
CHKHLT  033136          7284#   7296
CHKSCP  033146          7288#   7294
CHKSP   023266          5608#
CHKTYP  033102          7274#   7276
CLKSET  031302          6831    6946#
CLRTBI  002676          1570    1854#   5417    5561    5605    6960    7030
CLRO    010222          2795    2796    2797    2798    2800#
CMPB1   015644          4338    4340#
CMPB2   016506          4563    4564    4566#
```

```
CMPB3   017066          4690    4693#
CMPN    023522          5686#
CMPO    014770          4095    4096    4097    4099#
CMPOA   015232          4185    4188#
CMP1    015314          4214    4215    4216    4218#
CMP1A   015434          4259    4260    4261    4263#
CMP2    016142          4442    4443    4445#
CMP7    017632          4822    4823    4825#
CNTRLC= 000003          1992#   1998
CNTRLO= 000017          1993#   2011
CNVADR  003102          1923#   1986
CNVDAT  003056          1903#   1978    2029    6969
CNVDIG  003202          1909    1948#   1964
COMB1   011600          3206    3207    3209#
COMB1A  011760          3272    3273    3275#
COMB2   012326          3404    3405    3407#
COMB5   013062          3607    3609#
COMB6   013674          3807    3808    3810#
COMB7   014524          4012    4013    4015#
COMO    010266          2815    2816    2817    2818    2820#
COM1    011346          3117    3118    3120#
COM3    012740          3559    3561#
COM4    012056          3308    3310#
COM6    013270          3684    3685    3687#
COM7    014356          3958    3959    3961#
COUNT   004330          2136*   2139*   2142*   2160#
CPCHK   006260          2493#
CPERR   004030          2049    2106#
CPPASS  032652          6989    7213#
CPUERR= 177766          1138#   1448*   2050    2497*
CRLF    000752          1405#   2000    2007    2138    2226    2298    2550    2565    6852    7281
CYLADR  002300          1632    1657    1700*   1701*   1710#
DBINB7  020242          4946#   4978*   4979*   4983*   4984*   4985    4989*   4996    4997*   5004*   5006    5007*
DBIN7   017532          4797#   4812*   4816*   4821    4827*   4835*   4842*   4846*   4850
DDATA   017076          4699#   4703*   4705*   4709    4713    4714    4717*   4718    4723*   4730*   4737*   4738    4741*
                        4744    4748
DDATAB  017526          4767*   4768*   4773*   4776    4780*   4781    4785    4792#
DEC81   011550          3192    3193    3195#
DEC81A  011700          3244    3245    3247#
DEC82   012062          3457    3458    3460#
DEC85   013222          3665    3667#
DEC86A  013774          3838    3839    3840    3842#
DEC87   014572          4031    4032    4034#
DECO    010346          2841    2842    2843    2844    2846#
DEC1    011140          3040    3042#
DEC1A   011410          3138    3140#
DEC2    012224          3368    3369    3371#
DEC5    012714          3545    3546    3547    3549#
DEC6    013436          3733    3734    3736#
DEC7    014250          3923    3924    3926#
DEV     000757          1407#   1498*   1513*   1573*   1574    1599*   1601    1615    1618    7102*   7138*
DEVCHK  006544          2547    2553#
DEVERR  000742          1403#   2040*   2041
DEVICE  004422          1603    2040    2174#
DEVID   000772          1426#   1511*   1514*   1615*   1695    1843*   2037    2405*   2484*   7136*   7139*
DEVNAM  032764          1603*   7246#
```

```
DEVSTA  006560          2556* 2588  2594  2599  2605  2610  2613  2618  2621
DEVTBL  002352          1577  1734*
DIGBUF  003256          1905  1925  1969* 1987
DIGITS  003260          1970* 1979  2030  2031  6970
DIGTAB  003770          2093*
DISPLA= 177570          1154* 2375* 7291*
DISPLY  005436          2364* 2370* 2371* 2374* 2375  2633  2741  3878  4899  5408  5752  6162  6577
                        7081
DIVO    026152          6319*
DONE    032542          6986  7176* 7177
DONE1   032626          7189  7194*
DSCAN   005312          2287  2331*
DSKADR  002202          1631  1690*
EABITS  000762          1411* 1499* 1616  7119*
ECHO    000740          1402* 2017* 2018  2230* 2231
EISOPT= 040000          1281* 2516  6168
EMTVEC= 000030          1114* 2480* 5311* 5313* 5350* 5351* 5356* 6513* 6532*
EMT1    021744          5310  5318*
EMT1B   022024          5318  5322  5323  5326  5327  5331  5332  5335  5336  5337  5342*
EMT1C   022030          5316  5344*
EMT1D   022042          5345  5346  5349*
END     031352          6957*
ENDCP   027326          6494  6561*
ENDM    032502          7014  7018  7160*
ENDMEM  032500          7082  7159*
ENDMSG  032741          7187  7241*
ENDNS   033004          6981  7252*
ENDSIZ  007032          2567  2623*
ENDTAG  033010          2443  7081* 7089  7255*
END1    031356          6958* 7172
ERFLAG  005566          2079  2081* 2391  2402* 2483*
ERMSG   005604          2400  2408*
ERPRT   005540          1590  2304  2390* 2472  2542  2559  5063  5507  6548  6650  6958  7093
ERREG   004036          2052  2107*
ERRPRT  005560          2383  2387  2401*
ERRREG= 177744          1139* 1449* 2053
ERRVEC= 000004          1107* 1347* 1587* 1590* 2263* 2304* 2432* 2433* 2472* 2493* 2542* 2553* 2559*
                        5021* 5022* 5023  5063* 5421* 5422* 5464* 5507* 5508* 6539* 6548* 6611* 6612*
                        6650* 6651* 6958* 7082* 7093*
ERTAG   005572          2380* 2386* 2400* 2404*
EXTINS  023730          5759*
FACTOR  001004          1436* 1457  1483  1491* 2059  2062  2636* 2744* 3754  3881* 4701  4759  4868
                        4869  4873  4874  4902* 4967  5022  5146  5158  5166  5181  5205  5411* 5513
                        5518  5529  5755* 5858  6165* 6455  6472  6580* 6827*
FISOPT= 010000          1283* 2525
FPEVEC= 000244          1128*
FPOPT = 020000          1282* 2521
FRSTAD  001010          1440* 1478  2371  2630* 2738* 3875* 4507  4896* 5405* 5749* 6159* 6574*
FRSTME  001012          1441* 1486  2443* 2452*
GDAOR   004332          2145  2162*
GDDAT   005341          2292  2335*
GSTST   010740          2985*
HALT1   027226          6537*
HLT   = 104400          1318* 1503  1528  1646  1675  2756  2764  2772  2779  2787  2800  2810  2820
                        2828  2837  2846  2855  2864  2873  2879  2886  2893  2901  2923  2933  2943
                        2950  2957  2966  2982  3017  3035  3042  3051  3058  3064  3071  3078  3086
```

		3093	3101	3106	3114	3120	3128	3135	3140	3158	3164	3170	3176	3182
		3188	3195	3202	3209	3217	3224	3228	3235	3240	3247	3254	3260	3264
		3269	3275	3279	3285	3288	3305	3310	3317	3324	3330	3335	3339	3347
		3354	3361	3365	3371	3375	3381	3385	3388	3407	3411	3415	3423	3431
		3438	3446	3454	3460	3466	3472	3477	3482	3488	3496	3500	3503	3521
		3528	3534	3541	3549	3555	3561	3566	3572	3579	3595	3609	3613	3620
		3625	3633	3639	3644	3651	3655	3661	3667	3681	3687	3692	3700	3708
		3714	3722	3730	3736	3742	3746	3749	3763	3768	3774	3783	3790	3797
		3804	3810	3817	3823	3829	3835	3842	3847	3853	3859	3904	3911	3919
		3926	3933	3941	3948	3954	3961	3965	3972	3977	3984	4001	4009	4015
		4022	4028	4034	4040	4047	4053	4060	4067	4076	4084	4090	4099	4107
		4122	4127	4135	4145	4153	4161	4164	4174	4179	4188	4209	4218	4226
		4235	4242	4249	4255	4263	4274	4281	4286	4293	4301	4306	4329	4340
		4351	4356	4362	4367	4373	4425	4438	4445	4450	4457	4468	4478	4485
		4491	4499	4506	4546	4556	4560	4566	4573	4578	4607	4615	4620	4623
		4629	4634	4639	4671	4674	4680	4687	4693	4711	4716	4721	4727	4733
		4740	4746	4750	4771	4775	4778	4783	4787	4815	4819	4825	4832	4840
		4844	4848	4852	4911	4917	4922	4926	4932	4937	4976	4981	4987	4991
		5001	5011	5027	5029	5031	5033	5035	5037	5039	5041	5043	5045	5047
		5049	5051	5053	5055	5081	5101	5117	5135	5143	5154	5179	5203	5213
		5225	5233	5237	5259	5299	5317	5342	5349	5355#	5367	5369	5372	5376
		5387#	5458	5495	5524	5525	5526	5567	5586	5591	5617	5631	5638	5646
		5652	5663	5681	5696	5705	5768	5781	5795	5809	5823	5840	5855	5864
		5875	5881	5886	5897	5903	5908	5920	5939	5956	5960	5965	5971	5975
		5981	5990	5996	6017	6021	6025	6071	6076	6116	6119	6130	6142	6195
		6218	6238	6262	6293	6312	6340	6366	6390	6416	6430	6463	6468	6517
		6529	6542	6559	6597	6625	6645	6693	6733	6782	6809	6914	7132	7285
ICNT	001000	1429#	2028	2363	2460*	6968	6982*	6984	6987					
INCB1	011464	3161	3162	3164#										
INCB2	012556	3492	3493	3494	3496#									
INCB3	013114	3623	3625#											
INCB6	013606	3781	3783#											
INCB6A	013756	3832	3833	3835#										
INCB7	014556	4025	4026	4028#										
INCO	010366	2850	2851	2852	2853	2855#								
INC1	011254	3082	3083	3084	3086#									
INC3	012750	3564	3566#											
INC4	012106	3321	3322	3324#										
INC6	013400	3718	3719	3720	3722#									
INC7	014416	3975	3977#											
IODEV	001430	1500	1570#	7120										
IODONE	001426	1531#	1607*	1640*	1643*	1669*	1672*							
IORETR	000760	1408#	1608*	1649*	1654*	1677*								
IOTTST	021502	5245#												
IOTVEC=	000020	1112#	1322	2448*	2478*	2479*	5245							
ITCNT	001150	1464*	1466*	1469#	2464*	2467*	6992*	6996*						
JMP1	020762	5077	5078	5079	5081#									
JMP3	021032	5097	5101#											
JMP4	021070	5113	5117#											
JMP5	021134	5131	5135#											
JMP6	021154	5143#												
JMP7	021212	5154#												
JSRTST	021216	5158#												
JSR1	021300	5165	5167	5170	5171	5172	5173	5179#						
JSR1A	021302	5168	5181#											
JSR3	021366	5192	5199	5203#										

```
JSR3A   021370          5193    5205*
JSR4    021444          5215    5216    5217    5218    5222    5225*
JSR4A   021446          5214    5228*
JSR6A   021500          5232    5240*
KDPAR0= 172360          1270*   6752
KDPDR0= 172320          1267*   6744
KIPAR0= 172340          1239*   1937    2252*   6749    7041*   7142*
KIPAR1= 172342          1240*   2254*   7042*   7143*
KIPAR2= 172344          1241*   2256*   2313*   2374    7043*   7044    7106    7142
KIPAR3= 172346          1242*   7044*   7045*   7143
KIPAR4= 172350          1243*   6762*
KIPAR7= 172356          1244*   2259*   7046*   7063    7074
KIPDR0= 172300          1233*   2253*   6741    7035*
KIPDR1= 172302          1234*   2255*   7036*
KIPDR2= 172304          1235*   2257*   7037*
KIPDR3= 172306          1236*   2258*   7038*
KIPDR4= 172310          1237*   6763*
KIPDR7= 172316          1238*   2260*   7039*
KJOPT = 004000          1284*   2529    6585
KM    = 000000          1098*
KPTR  = 000600          1276*   1379    2422    2447    2459    5498    6961    7149
KSP   =%000006          1082*
KTABRT  005510          2380*   2477    6531
KTABT   030176          6659    6739    6760*
KTAMSG  005623          2380    2411*
KTEX    030452          6761    6817*
KTOPT = 100000          1280*   2511
KTPAR   030012          6701*
KTPDR   027646          6658*
KW11    030666          6842    6865*
LDDISP  005432          2363*   2485    2632    2740    3877    4898    5407    5751    6161    6576    7080
LKOPT = 001000          1286*   2537    6875    6951
LKS   = 177546          1140*   2535    6878*   6879    6928*   6931*   6955*
LKSRV   031176          6866    6922    6927*   6946    6953
LKVEC = 000100          1120*   6866*   6868*   6908*   6909    6922*   6946*   6953*   6954*
LOGICA  032616          1332    2428    7190*
LPB   = 177516          1156*
LPS   = 177514          1155*
LSTMEM  005750          1488    2439*   2442*   2456*
LTICKS  000774          1427*   6927*
MARKEX  024736          6018    6022    6026*
MARK1   024712          6014    6016*
MEMTBL  000730          1398*   1845    2132    2203
MMON    000770          1424*   1476    1522    1929    2057    2372    2461*   5557    6501    6671    6710    6760    7079*
                        7171*
MMVEC = 000250          1129*   2264*   2305*   2477*   6497*   6531*   6770    6771    6774*   6775*   6813*   6814*
MOVB1   015620          4323    4324    4325    4326    4329*
MOVD    014720          4073    4074    4076*
MOVDA   014750          4088    4090*
MOV1    015504          4284    4286*
MOV7    017602          4813    4815*
MPI     027020          6170    6400    6493*
MRKTST  024654          6003*
MSG     030656          6836*   6837    6846    6854*   6856    6859*   6977*
MSG1    032710          2449    7235*
MSG2    032725          2453    7238*
```

```
MULO    026034  6272#
N     = 000010  1090#
NEGB1   011534  3185    3186    3188#
NEGB4   012506  3469    3470    3472#
NEGB6   013712  3814    3815    3817#
NEGB7   014640  4050    4051    4053#
NEGO    010406  2860    2861    2862    2864#
NEG1    011400  3132    3133    3135#
NEG2    012130  3333    3335#
NEG5    012726  3552    3553    3555#
NEG6    013324  3696    3697    3698    3700#
NEG7    014324  3945    3946    3948#
NEXPAR  032432  2463*   7043    7060    7071    7141#   7170*
NODEV   032762  1604    7245#
NOTFND  005235  2286    2308    2322#
NOTHOM  007024  2570    2592    2597    2603    2608    2616    2620#
NOTYPE  006256  1872    2013*   2489#
NULL  = 000000  1898#
NULLS   030662  6836    6861#
OAERR   020526  5014#
OLDPSW  031676  7028#   7151
OPTDEV  033066  2563    7268#
OPTION  000770  1420#
OPT.CP  000766  1370    1413#   1446    1870    2047    2249    2272    2544*   5396    5514    5568    5571    6091
                6168    6398    6505    6585    6658    6663    6704    6766    6795    6829    6870    6875    6951
                6962    6983    7004    7052    7066    7161
OVTLW   022306  5415#
PARCSR= 172100  1342#   1348
PAREVR  000707  1390#   2244
PARTBL  030160  6701    6749#
PARVEC= 000114  1343#   1345*   1346*   2262*   2303*
PASCNT  004000  2032    2097#
PASSES  004010  2030*   2031*   2102#   6971
PASSNO  004164  2125#   6972
PASTAB  032676  6984    7228#
PC    = 000007  1065#   1356*   1359*   1360*   1361*   1380*   1500*   1507*   1515*   1516*   1520    1530    1570*
                1581*   1584*   1609*   1616*   1631*   1683*   1708*   1709*   1711*   1713*   1851*   1855*   1860*
                1877*   1882*   1890*   1896*   1904*   1924*   1965*   1966*   1978*   1980*   1986*   1988*   2027*
                2029*   2036*   2042*   2046*   2051*   2054*   2063*   2067*   2069*   2135*   2143*   2148*   2151*
                2155*   2158*   2159*   2270*   2289*   2293*   2296*   2349*   2360*   2363*   2365    2376*   2401*
                2438*   2462*   2485*   2488*   2545    2549*   2560    2564*   2628    2632*   2634    2637    2727
                2729*   2736    2740*   2742    2745    2988*   2990*   3002*   3026    3146    3294    3394    3510
                3593    3671*   3673    3864    3866*   3873    3877*   3879    3882    3890    3992    4072    4175
                4177    4181    4184    4196    4313    4378    4502    4515    4587    4651    4801    4859*   4860
                4861    4862*   4863    4864*   4865    4866    4867*   4868    4869    4873*   4874*   4876    4877
                4878    4879*   4880*   4885    4887*   4894    4898*   4900    4903    4949    5015    5069    5084
                5104    5119    5207    5230    5231*   5239*   5246    5265    5309    5357    5363    5389    5391*
                5395    5403    5407*   5409    5412    5417*   5419    5423*   5472    5476*   5486*   5501    5503
                5522*   5561*   5596    5598    5605*   5666    5672*   5674    5686    5737    5739*   5747    5751*
                5753    5756    5770    5783    5811    5817    5818    5834    5835    5866    6004    6007    6008
                6009    6010    6011    6014*   6053    6061    6117    6125    6136    6147    6149*   6157    6161*
                6163    6166    6177    6180*   6203    6206*   6226    6247    6274    6275*   6298    6300*   6320
                6354    6357*   6378    6381*   6403    6423    6495    6511    6537    6544    6563    6565*   6572
                6576*   6578    6581    6609    6613    6635    6772    6777    6810    6819    6821*   6828    6851*
                6858*   6865    6898*   6903    6957*   6960*   6969*   7006*   7027*   7030*   7033    7080*   7120*
                7131*   7140*   7190*   7280*   7290
```

```
FDRTBL  030142          6660    6741#
PDWN    000610          1325    1370#   1377    2475
PEFLG   000761          1409#   2084    2245#   2247#   2486#
PERET   004646          2086    2247#
PFMIL   000666          1384    1387#
PFVEC = 000024          1113#   1373#   1377#   2475#   2476#
PHYSPC  004047          2065    2109#
PIRQ  = 177772          1136#   2500    6448
PIRQQ   026604          6444#
PIRVEC= 000240          1127#   6450#   6451#   6454#   6455#   6459#   6485#   6486#
PIR4  = 010000          1315#
PKM   = 000000          1101#
PLKCSB= 172542          1207#   6872#   6936#
PLKCSR= 172540          1206#   2531    6873#   6904    6937#   6940#
PLKCTR= 172544          1208#
PLKOPT= 002000          1285#   2533    6870
PLKSRV  031226          6867    6923    6924    6935#   6947
PLKVEC= 000104          1121#   6867#   6869#   6909#   6923#   6947#
PNTREG  004172          2042    2130#
PPB   = 177556          1144#
PPOPT = 000002          1300#
PPS   = 177554          1143#
PPVEC = 000074          1119#
PRB   = 177552          1142#
PRDAT   000772          1421#
PROPT = 000001          1299#
PRS   = 177550          1141#
PRSYNC  000773          1422#
PRTY2 = 000100          1097#   6109    6117
PRTY3 = 000140          1096#   5255
PRTY4 = 000200          1095#   1328    1337    1572    5250    5263    6134    6140    7029
PRTY5 = 000240          1094#   1630    6107
PRTY6 = 000300          1093#
PRTY7 = 000340          1092#   1324    1326    1330    2390    2479    5254    5302    5382    5418    6058    6060    6066
                        6123    6128    6135    6406    6451    6453    6849
PRVEC = 000070          1118#
PSM   = 010000          1102#   5279
PSW   = 177776          1134#   1452#   1572#   1854    1862#   2390#   2503    2985    3000#   3005#   5303#   5313    5361
                        5415    5416#   5418#   5446#   5451#   5559    5560#   5563    5603    5604#   5664#   6050    6058#
                        6066#   6107#   6113    6123#   6128    6134#   6140    6144#   6180    6184    6188#   6190    6206
                        6212    6357    6361    6381    6385    6450    6456#   6487#   6493    6530#   6554    6557    6560#
                        6612    6619    6775    6785    6849#   6850    6887    6893#   6910#   6959#   7027    7029#   7150#
PSWBIT  032664          5569    7221#
PSWCHK  023054          5557#
PSWTAB  032632          6993    7202#
PTICKS  000776          1428#   6935#
PUM   = 030000          1103#   5278    5284    6107    6123    6128    6134    6140
PUP     000632          1373    1377#
QV      000771          1425#   1492    1595    1690    2425#   2430#   2440    2465    7096
RC      006750          2579    2601#
RCCA  = 177452          1178#   1776
RCCS  = 177446          1176#   1778
RCDA  = 177442          1175#   1774    1775
RCLA  = 177440          1174#   2199
RCOPT = 000010          1293#   2604
RCTBL   002460          1738    1773#
```

```
RCVEC = 000210          1124#   1773
RCWC  = 177450          1177#   1777
RECO    004510          2209#   2450    2454
REG   = 004000          1104#   6096    6099    6107    6128    6140
REGADR  004466          2131    2195#
REGS    004444          2130    2185#
RELOC   001152          1472#   2729    3866    4887    5391    5739    6149    6565    6821
RELPC   004042          2061    2108#
RELR1   001006          1438#   1456*   1457*
REL0    007032          2628#
REL00   010030          2633*   2730#
REL1    010032          2736#
REL11   014060          2741*   3867#
REL2    014062          3873#
REL22   020056          3878*   4888#
REL3    020060          4894#
REL33   022214          4899*   5392#
REL4    022242          5397    5403#
REL44   023662          5408*   5740#
REL5    023664          5747#
REL55   025376          5752*   6150#
REL6    025400          6157#
REL66   027342          6162*   6566#
REL7    027344          6572#
REL77   030464          6577*   6822#
RESERR  005530          2386#   2473    2543    5064    5552    6549
RESET1  027300          6553#
RESMSG  005640          2386    2414#
RESPSW  002712          1858#   1864
RESTPS  002722          1581    1683    1862#
RESTRP  022702          5512#
RESVEC= 000010          1108#   2473*   2474*   2494*   2543*   5023*   5064*   5517*   5518*   5552*   6540*   6549*
RETPC   001366          1485*   1517#
RETPSW  002704          1856#   1863
RETRY   032216          7086#   7137
RF      006714          2575    2590#
RFCMA = 177464          1169#   1756
RFDAE = 177470          1171#   1754
RFDAR = 177466          1170#   1755
RFDCS = 177460          1167#   1758    2197
RFOPT = 000002          1291#   2593
RFTBL   002414          1736    1753#
RFVEC = 000204          1122#   1753
RFWC  = 177462          1168#   1757
RHCS2 = 000000          1194#
RK      006706          2573    2587#
RKBA  = 177410          1163#   1746
RKCS  = 177404          1161#   1748
RKDA  = 177412          1164#   1744    1745
RKDS  = 177400          1159#   2196
RKER  = 177402          1160#
RKOPT = 000001          1290#   2587
RKTBL   002372          1735    1743#
RKVEC = 000220          1125#   1713
RKWC  = 177406          1162#   1747
RK6     007010          2585    2615#
```

DCQKCG.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
RK6BA = 177444          1224#   1824
RK6CS1= 177440          1222#   1817    1826
RK6CS2= 177450          1226#   1820
RK6DA = 177446          1225#   1823
RK6DC = 177460          1230#   1822
RK6DS = 177452          1227#
RK6ER = 177454          1228#
RK6OF = 177456          1229#   1818
RK6UT= 000100           1296#   2617
RK6TBL  002572          1741    1816#
RK6VEC= 000210          1126#   1821
RK6WC = 177442          1223#   1825
ROLB1   011476          3167    3168    3170#
ROLB2   012400          3427    3428    3429    3431#
ROLB3   013170          3648    3649    3651#
ROLB6   013726          3820    3821    3823#
ROLB6A  014022          3850    3851    3853#
ROLB7   014656          4057    4058    4060#
ROLO    010440          2876    2877    2879#
ROL1    011174          3054    3055    3056    3058#
ROL1A   011206          3061    3062    3064#
ROL3    012776          3576    3577    3579#
ROL4    012204          3358    3359    3361#
ROL6    013254          3677    3678    3679    3681#
ROL7    014434          3981    3982    3984#
RORB1   011564          3199    3200    3202#
RORB1A  011654          3232    3233    3235#
RORB4   012362          3419    3420    3421    3423#
RORB5   013132          3629    3630    3631    3633#
RORB6   013660          3801    3802    3804#
RORB7   014542          4019    4020    4022#
RORO    010324          2832    2833    2834    2835    2837#
RORI    011124          3032    3033    3035#
RORIA   011236          3074    3075    3076    3078#
ROR2    012072          3314    3315    3317#
ROR5    012650          3525    3526    3528#
ROR6    013304          3704    3705    3706    3708#
ROR7    014306          3938    3939    3941#
RP      006732          2577    2596#
RPBA  = 176720          1201#   1766
RPCA  = 176722          1202#   1764
RPCS  = 176714          1199#   1768
RPDA  = 176724          1203#   1765
RPDS  = 176710          1197#   2198
RPER  = 176712          1198#
RPOPT = 000004          1292#   2598
RPTBL   002436          1737    1763#
RPVEC = 000254          1130#   1763
RPWC  = 176716          1200#   1767
RP4     006766          2581    2607#
RP4AS = 176716          1188#
RP4BA = 176704          1183#   1792
RP4CA = 176734          1191#   1790
RP4CS1= 176700          1181#   1785    1794    2200
RP4CS2= 176710          1185#   1788
RP4DST= 176706          1184#   1791
```

```
RP4DS1= 176712           1186#
RP4ER1= 176714           1187#
RP4LA = 176720           1189#
RP4OF = 176732           1190#   1786
RP4OPT= 000020           1294#   2609
RP4TBL  002502           1739    1784#
RP4VEC= 000254           1131#   1789
RP4WC = 176702           1182#   1793
RS      007002           2583    2612#
RSAS  = 172056           1218#
RSBA  = 172044           1213#   1808
RSCS1 = 172040           1211#   1801    1802    1810    2201
RSCS2 = 172050           1215#   1804
RSDA  = 172046           1214#   1806    1807
RSDS  = 172052           1216#
RSER  = 172054           1217#
RSLA  = 172060           1219#
RSOPT = 000040           1295#   2612
RSTBL   002536           1740    1800#
RSVEC = 000204           1123#   1805
RSWC  = 172042           1212#   1809
RTI1    031620           5398*   6997    6999#
RTT1    024742           6050#
RTT1EX  025132           6086#
RTT2    025140           6088    6091#
RTT2A   025154           6092    6094#
RTT2EX  025354           6052    6093    6103    6118    6120    6129    6131    6141    6144#
R0    =%000000           1058#   1348*   1351*   1478*   1479    1502    1503    1505    1509    1635    1681*   1694    1695*
                         1696*   1697*   1701    1705    1706    1707*   1846    1847    1867    1868*   1875    1879*   1908*
                         1939*   1942*   1946*   1963*   2130*   2140*   2209    2213*   2214*   2215    2224    2230    2232*
                         2236    2434*   2435    2437*   2438    2495*   2498*   2501*   2504*   2506*   2507    2518    2523*
                         2628*   2629    2630    2634*   2635*   2636    2736*   2737    2738    2742*   2743*   2744    2794*
                         2804    2814*   2823*   2831*   2840*   2849*   2859*   2867*   2875*   2881*   2889*   2895*   2896*
                         2905*   2907*   2908    2920    2928*   2929*   2931*   2935*   2939*   2994*   2995    3006    3011
                         3396*   3397    3399*   3513*   3515    3596*   3597*   3600    3601*   3602    3603*   3673*   3674
                         3747    3753*   3754*   3873*   3874    3875    3879*   3880*   3881    3890*   3891    3892    3893*
                         3894    3895*   3896    3897    3898*   3899*   3988    3989*   3990    3991*   4072*   4078    4080
                         4101    4131    4137    4140    4147    4149    4151    4155    4158    4175*   4176    4177    4181*
                         4182*   4183    4184    4186    4519*   4520    4526    4529    4531    4534    4536*   4539    4541*
                         4544    4549    4588*   4589    4590    4592    4594    4595    4596    4597    4651*   4652    4653
                         4656    4658    4660    4661    4663    4735*   4736    4738*   4751    4801*   4802    4803    4805
                         4807    4808    4857*   4862    4865    4866    4894*   4895    4896    4900*   4901*   4902    4949*
                         4950    4951*   4952*   4953*   4954    4955*   4956*   4957*   4965*   4966*   5069*   5070*   5072
                         5089*   5091*   5098    5099    5105*   5114    5115    5125*   5132    5133    5182*   5191*   5194*
                         5230*   5234    5235    5249*   5250*   5252    5254*   5255*   5256    5257*   5261    5290*   5293*
                         5297    5308*   5319*   5320*   5321*   5325*   5330*   5334*   5339*   5344*   5347    5363*   5373*
                         5374    5403*   5404    5405    5409*   5410*   5411    5455*   5467*   5482*   5485*   5488*   5493
                         5568*   5569*   5573*   5575    5622*   5623    5636    5650    5661    5674*   5675*   5676*   5677*
                         5686*   5687    5691    5700    5747*   5748    5749    5753*   5754*   5755    5759*   5761*   5766*
                         5770*   5771    5775    5783*   5784    5787    5788*   5790    5793    5797*   5798    5800*   5805
                         5818*   5819    5820*   5821    5848*   5859    5926*   5932*   5937    5942*   5947*   5948    5958*
                         5961*   5963    6005*   6023    6059*   6064*   6069    6073    6080*   6094*   6095*   6097    6100
                         6102    6113*   6117    6157*   6158    6159    6163*   6164*   6165    6173*   6176    6196*   6197
                         6200*   6202    6219*   6220    6225*   6230    6233*   6246*   6252    6254*   6257*   6272*   6283
                         6295*   6306    6310    6319*   6324    6329    6335    6344*   6351*   6360    6368*   6370    6374*
                         6384    6392*   6394    6444*   6466    6554*   6557    6560    6572*   6573    6574    6578*   6579*
```

		6580 6593* 6594 6595 6600* 6630* 6668* 6673* 6676* 6679 6683* 6684 6688*
		6707* 6712* 6717* 6718 6722* 6723 6728* 6776* 6780 6785* 6786* 6787* 6788*
		6789* 6790 6792* 6793 6797* 6798 6800* 6801 6803* 6804 6805 6903* 6983*
		6984 7034* 7035 7036 7037 7038 7039 7057 7058 7059 7068 7069 7070
		7087* 7092 7123 7129 7134 7181* 7182* 7289* 7291 7292*
R1	=%000001	1059* 1455 1456 1467* 1576* 1593 1698 1702 1906* 1926* 1927 1931* 1932*
		1933* 1934* 1935* 1936* 1937* 1938* 1940* 1944* 1951* 2066* 2090 2146* 2147
		2153* 2154 2248* 2266 2276 2277* 2278 2281* 2282 2295 2314* 2637* 2745*
		2987 2995* 2996 3012 3020* 3882* 3992* 4587* 4588 4903* 5395* 5412* 5756*
		6166* 6581* 6686 6725 6828* 6865* 7033* 7290*
R10	=%000000	1066*
R11	=%000001	1067*
R12	=%000002	1068* 6108*
R13	=%000003	1069*
R14	=%000004	1070*
R15	=%000005	1071*
R2	=%000002	1060* 1349* 1354* 1480 1482 1435 1486* 1487 1490 1502* 1505 1515 1660
		1682* 1808 1849 1906 1949* 1952* 1957* 1961* 1962 2028* 2033* 2034 2037*
		2039* 2040 2045* 2050* 2053* 2055* 2056 2062* 2134* 2150* 2157* 2251* 2253
		2255 2257 2260 2275* 2277 2278 2280* 2281 2282 2284* 2288* 2295* 2507*
		2511* 2516* 2521* 2525* 2529* 2533* 2537* 2541* 2544 2555* 2587* 2593* 2598*
		2604* 2609* 2612* 2617* 2727* 2728* 2908* 2909* 2910 2919 2936* 2938* 2941*
		2969* 2970* 2971 2973* 2974* 2976* 2996* 2997 3013 3026* 3027* 3028* 3031*
		3039* 3046* 3053* 3060* 3066* 3073* 3081* 3088* 3096* 3104* 3109* 3116* 3123
		3131* 3137* 3397* 3398* 3409* 3426* 3440* 3442* 3456* 3468* 3479 3485* 3498
		3501 3593* 3594 3595 3596 3598 3599 3600* 3602* 3603 3604 3606* 3611
		3622* 3635* 3641* 3653* 3663* 3664* 3864* 3865* 3894* 3895 3898 3899 4078*
		4080* 4087 4094 4101* 4102 4103 4104* 4105 4111* 4112* 4114* 4116 4118
		4123 4125 4129* 4131* 4183* 4186 4316* 4317* 4318* 4319* 4322 4331* 4333*
		4337 4347 4353 4358 4364* 4365 4369* 4371 4385* 4386* 4392* 4393 4394*
		4395 4396* 4398* 4400 4401 4412* 4413 4419 4431* 4434 4447 4453 4459*
		4463 4470* 4471 4474 4481 4489 4493* 4494* 4495* 4502* 4503 4504 4507
		4520* 4521* 4529 4539 4593* 4597* 4603 4609* 4611 4618 4621 4624* 4626
		4631 4637 4701* 4702 4759* 4760 4803* 4804 4805* 4806 4885* 4886* 4959*
		4968* 5014* 5026* 5028* 5030* 5032 5034* 5036* 5038* 5040* 5042* 5044* 5052
		6083* 5094* 5096* 5103* 5107* 5112* 5127* 5130* 5138* 5141* 5148* 5152* 5159*
		5160* 5163 5164 5174* 5185* 5200 5201 5219* 5220* 5221 5261* 5274 5280
		5285 5288* 5381* 5382* 5383* 5384* 5385* 5389* 5390* 5456* 5466* 5469* 5471*
		5473* 5475* 5477* 5482 5512* 5513* 5519 5521 5574* 5575 5578 5582 5584
		5588 5592* 5615* 5656* 5657 5659 5687* 5688* 5691 5700 5737* 5738* 5771*
		5773* 5779 5789* 5792* 5793 5798* 5800 5807* 5811* 5812 5815* 5816* 5817*
		5821* 5857* 5859* 5890* 5891 5941* 5948 5951* 5954 6003* 6019 6026 6096*
		6097* 6099* 6100* 6106* 6114 6147* 6148* 6176* 6179 6189* 6202* 6205 6209*
		6211* 6229* 6232* 6234 6247* 6248 6252* 6258 6274* 6275 6279 6283* 6289
		6298* 6299* 6300 6303 6306* 6308 6322* 6324* 6334 6352* 6370 6375* 6394
		6403* 6421 6423* 6435 6446* 6457 6460* 6461 6470 6563* 6564* 6587* 6588*
		6589 6591* 6594* 6595 6602* 6613* 6615 6634* 6642* 6647* 6649* 6660* 6665*
		6666* 6668 6701* 6707 6777* 6779 6819* 6820* 6968* 6970* 6973 6974 6975
		6976 6987* 6988* 6989* 7086* 7090 7106* 7107* 7108* 7109* 7110* 7111* 7113*
		7122* 7123* 7125 7127* 7129 7188* 7190 7279*
R3	=%000003	1061* 1482* 1503 1907* 1928* 1941* 1945* 1950* 2131* 2132 2134 2146 2150
		2153 2157 2276* 2288 2910* 2911* 2912 2918 2945* 2946* 2948* 2971* 2975
		2978* 2997* 2998 3014 3146* 3147* 3148 3150* 3153* 3178* 3184* 3191* 3198*
		3211* 3243* 3250* 3265* 3271* 3277* 3281* 3283* 3286 3510* 3511* 3512* 3513
		3514 3515* 3516 3519 3524* 3537* 3551* 3563* 3581* 4087* 4094 4102* 4103*
		4105 4137* 4140* 4157* 4158* 4162* 4165 4166* 4169 4199* 4200* 4202* 4205

```
                    4213   4222   4231   4237*  4238*  4240   4245   4251   4258   4265*  4267*  4268   4270
                    4277   4283   4288   4295*  4296   4297   4342*  4344*  4345   4383*  4384*  4385   4395*
                    4397   4399*  4403   4404   4406*  4407   4409*  4410   4421   4422*  4423   4515*  4516
                    4519   4534   4544   4549*  4550*  4551*  4552   4558   4562   4569   4576   4596*  4598*
                    4653*  4654   4655   4656*  4657*  4658*  4659   4668*  4669*  4675   4676*  4677*  4681*
                    4682*  4683   4685   4689   4691   4761*  4762*  4763   4806*  4807   4808*  4906*  4907
                    4908   4914   4934   4960*  4969*  5015*  5018   5019*  5020*  5044   5046   5048   5050*
                    5084*  5087   5088*  5089   5090*  5091   5092   5099   5119*  5122   5123*  5124*  5125
                    5129   5133   5196*  5197*  5198   5207*  5210   5212   5256*  5257   5269   5271   5429*
                    5430*  5431   5465   5519*  5522   5531   5533*  5583*  5584   5589*  5608*  5616   5619
                    5623*  5625*  5627   5629   5634   5640*  5641*  5642   5644   5772*  5775*  5779   5858*
                    5866*  5870   5871   5879*  5884*  5889*  5891*  5899   5901*  5905*  5906   5977*  5978*
                    5979   6174*  6197   6201*  6220   6227*  6231*  6236   6249*  6260   6277*  6291   6302*
                    6310   6321*  6336   6354*  6355   6356*  6362   6407*  6408*  6414   6419*  6425*  6428
                    6433*  6447*  6456   6469*  6499*  6500   6616*  6617   6632   6643*  6646*  6675*  6687*
                    6714*  6726*  6778*  6779*  6971*  6973*  6975*  7090*  7091*  7092*  7125   7127   7128*
                    7134
R4      =%000004    1062*  1480*  1481*  1487*  1488   1490*  1520*  1524*  1525*  1526   1529*  1609   1611
                    1644*  1673*  1844   1845*  1846*  1847*  1848*  1849*  1850*  1905*  1925*  1962*  2512*
                    2514*  2912*  2913*  2914   2917   2952*  2953*  2955   2975*  2980*  2998*  2999   3015
                    3148*  3149*  3155*  3160*  3166*  3172*  3205*  3213*  3219*  3220*  3226*  3231*  3238*
                    3256   3262*  3294*  3295*  3296   3313*  3320*  3327*  3337*  3350*  3357*  3373*  3376*
                    3383   3386   3516*  3531*  3544*  3558*  3568*  3575*  3582*  4147*  4148*  4149*  4150*
                    4151   4155*  4156*  4157   4165*  4168*  4169*  4196*  4197   4198*  4199   4203*  4205*
                    4213   4222   4230*  4231*  4240*  4243*  4245*  4251*  4258   4268*  4269*  4270*  4277*
                    4283*  4288*  4296*  4297*  4303*  4304   4381*  4382*  4383   4393*  4403*  4404   4407
                    4413   4416   4418*  4419*  4421*  4429   4522*  4532   4542   4552*  4553*  4554*  4558*
                    4562   4569   4576*  4594*  4599*  4617*  4659*  4661*  4662*  4663*  4665*  4667*  4668
                    4669   4676   4677   4681   4682   4683   4685   4689   4691   4760*  4761   4907*  4908*
                    4909   4913*  4914*  4915   4919*  4920   4924   4928*  4929*  4930   4934*  4935*  4961*
                    4970*  5104*  5105   5109   5110   5111   5115   5175*  5176*  5177   5181*  5184   5187
                    5190*  5198   5205*  5206   5212*  5221   5223   5224*  5253*  5274   5278*  5279*  5280
                    5284*  5285   5302*  5303   5521*  5531   5563*  5565   5581*  5582*  5583   5588*  5589
                    5594*  5603*  5665   5825*  5827   5829*  5832   5871*  5873*  5877   5888*  5889   5894*
                    5895   5905   5906   5917*  5923*  5967*  5969*  5974*  5993*  5994   6178*  6183*  6191
                    6204*  6210*  6214   6228*  6230*  6234   6248*  6255*  6258   6278*  6281*  6287*  6289
                    6303*  6304*  6305*  6308   6334*  6335*  6355*  6360*  6362   6378*  6379   6380*  6386
                    6405*  6406*  6407   6414   6417   6428   6431   6448*  6449*  6457*  6461*  6466   6470*
                    6484*  6617*  6621   6632   6643   6646   6676*  6677*  6678   6679   6681*  6682*  6683
                    6684   6686*  6715*  6716*  6717   6718   6720*  6721*  6722   6723   6725*  6972*  6974*
                    6976*  7089*  7091
R5      =%000005    1063*  1479*  1481   1509   1571   1574*  1575*  1576   1577*  1582*  1589   1601*  1602*
                    1603   1624*  1625*  1626*  1627   1632*  1633*  1634   1635*  1636*  1637*  1644   1647*
                    1648*  1655*  1656*  1657*  1658*  1659   1660*  1661*  1663   1664*  1673   1676*  1680*
                    1927*  1943*  1944   1948*  1953*  1955*  1958*  2210*  2219*  2220*  2221*  2227   2228*
                    2233*  2234*  2235*  2236*  2450*  2454*  2554*  2556*  2557   2569   2572   2574   2576
                    2578   2580   2582   2584   2620*  2914*  2915*  2916   2959*  2960*  2962*  2964   2999*
                    3001   3016   3296*  3297*  3301   3307*  3332*  3342*  3344*  3363*  3367*  3386   3394*
                    3395*  3396   3403*  3412*  3418*  3434*  3449*  3462*  3474   3491*  3501   3604*  3616*
                    3628*  3647*  3658*  4116*  4118   4123*  4125   4313*  4314   4315*  4316   4322*  4327*
                    4334*  4335*  4337   4345*  4347   4353*  4358*  4365   4369   4378*  4379   4380*  4381
                    4397*  4400*  4401   4410   4415*  4416   4423*  4429*  4430*  4434*  4441   4447*  4453*
                    4460*  4463*  4471*  4472*  4474   4481*  4487*  4488*  4489   4503*  4504   4523*  4524
                    4527   4537   4590*  4591   4592*  4593   4603*  4610*  4611*  4618   4621   4626*  4631*
                    4636*  4637   4654*  4672   4678   4702*  4763*  4962*  4971*  5158*  5160   5163*  5166
                    5169*  5177   5183*  5184*  5185   5186*  5187*  5190   5201   5206*  5245*  5248*  5252*
```

```
                        5253    5263*   5264*   5914*   5918    5922*   5983    5992*   6004*   6005    6006    6013*   6016*
                        6023    6177*   6178    6179*   6191    6203*   6204    6205*   6214    6226*   6227    6236    6250*
                        6256*   6260    6279*   6291    6336*   6338    6379*   6384*   6386    6635*   6636*   6637    6661*
                        6667*   6677    6682    6702*   6706*   6716    6721
SAVPSW  025136           6050*   6051    6056    6081    6089*   6094    6144
SAVVAL  002646           1507    1843*   7131
SBCB1   011510           3173    3174    3176*
SBCB3   013210           3659    3661*
SBCB4   012474           3463    3464    3466*
SBCB6   013562           3771    3772    3774*
SBCB7   014470           3997    3998    3999    4001*
SBCD    010470           2890    2891    2893*
SBC1    011270           3089    3090    3091    3093*
SBC1A   011306           3097    3098    3099    3101*
SBC5    013010           3583    3585*
SBC6    013360           3711    3712    3714*
SBC7    014212           3908    3909    3911*
SBINB7  020232           4942*   4973*   4974    4978    4985    4989    4993*   4996*   4999    5003*   5006*   5009
SBIN7   017530           4796*   4811*   4812    4821    4827    4834*   4835    4842    4846    4850
SCOPE = 104000           1319*   2788    2902    3021    3141    3262    3389    3504    3586    3668    3750    3863    3985
                        4068    4189    4307    4426    4500    4509    4547    4579    4640    4694    4753    4788    4853
                        4882    4938    5012    5065    5155    5240    5305    5352    5386    5509    5553    5601    5669
                        5735    5909    5997    6027    6086    6145    6267    6313    6396    6437    6490    6534    6550
                        6561    6652    6698    6738    6815    6948    7293
SCOPEA  001014           1327    1446*   2480    5350    5385    6532
SDATA   017074           4698*   4704*   4705    4706*   4707*   4712*   4713*   4714    4718    4723    4729*   4730    4736*
                        4741    4742*   4743*   4744    4747*   4748    4751
SDATAB  017524           4765*   4766*   4767    4769    4773    4776    4780    4781    4791*
SDPAR0= 172260           1272*   6754
SDPDR0= 172220           1269*   6746
SECT    005442           2366*   2631*   2739*   3876*   4897*   5406*   5750*   6160*   6575*
SIPAR0= 172240           1262*   6507    6751    7071*   7072
SIPAR1= 172242           1263*   7072*   7073*
SIPAR4= 172250           1264*   6768*
SIPAR6= 172254           1265*   6507*
SIPAR7= 172256           1366*   7074*
SIPDR0= 172200           1257*   6743    7068*
SIPDR1= 172202           1258*   7069*
SIPDR4= 172210           1259*   6769*
SIPDR6= 172214           1260*   6508*
SIPDR7= 172216           1261*   7070*
SLASH   000755           1406*   2217
SLR     = 177774         1135*   2527*   6587
SM      = 040000         1099*   6099    6140
S080    024454           5915    5926*
S081    024462           5928*   5932
S0810   024436           5917*   5980
S082    024510           5928    5929    5930    5931    5933    5934    5935    5936    5939*
S083    024522           5943*   5951
S084    024554           5943    5944    5945    5946    5949    5952    5953    5956*
S085    024556           5958*
S085A   024576           5967*
S086    024622           5978*
S087    024630           5981*
S088    024632           5924    5983*
S089    024446           5922*   5993
```

```
SP     =%000006   1064* 1379* 1450  1455* 1467  1571* 1582  1611* 1612* 1613* 1614  1627* 1628*
                  1629* 1630* 1662* 1663* 1664  1680  1694* 1698* 1699* 1700  1702* 1703* 1704
                  1707  1844* 1850  1854* 1855  1857* 1858* 1863* 1867* 1868  1869* 1875* 1878
                  1879  1883  1886* 1888* 1895  1996* 1997* 1998  2001  2005  2008  2011  2014*
                  2017  2033  2045  2055  2066  2076* 2077* 2082  2088* 2209* 2245  2343* 2344*
                  2345* 2346* 2347* 2348* 2349  2353* 2354  2355  2356  2357  2358  2359  2365*
                  2367* 2368* 2369* 2370  2381* 2382* 2422* 2430  2447* 2459* 2471* 2916* 2917*
                  2918* 2919* 2920* 2921  2987* 2988  3001* 3002  3004* 3011* 3012* 3013* 3014*
                  3015* 3016* 3019* 3020  4522  4523  4526* 4527  4531  4532  4536  4537  4541
                  4542  4876  4877  4967* 4968  4969  4970  4971  5058* 5059* 5062* 5176  5197
                  5220  5235  5246* 5247* 5248  5265* 5266* 5267  5269  5304* 5309* 5310* 5311
                  5357* 5358* 5359  5374  5377  5378  5419* 5420* 5421  5425* 5426  5431* 5436
                  5437  5438* 5439* 5440* 5441* 5442* 5447* 5448* 5449* 5450* 5455  5456  5457*
                  5465* 5468* 5470* 5472* 5478* 5492* 5498* 5501* 5503* 5504* 5506* 5528* 5529*
                  5562* 5595  5596* 5598* 5599* 5608  5610* 5611* 5613* 5615  5616* 5619* 5622
                  5626* 5629  5633* 5636  5641  5644  5648  5650  5654* 5655* 5657  5659  5661
                  5665* 5666* 5667* 6003  6006* 6007* 6008* 6009* 6010* 6011* 6012* 6013  6019
                  6026* 6053* 6054* 6055  6056* 6057  6060* 6061* 6062* 6081* 6109* 6110* 6111*
                  6124* 6125* 6126* 6135* 6136* 6137* 6184* 6185  6188  6223* 6224* 6225  6239*
                  6240  6242  6244* 6245* 6246  6263* 6264  6266  6273* 6287  6294* 6350* 6356
                  6369* 6373* 6380  6393* 6471* 6472* 6495* 6496* 6497  6498* 6499  6500* 6509*
                  6511* 6512* 6513  6515* 6520  6521  6522  6525* 6526* 6533* 6537* 6538* 6539
                  6540  6544* 6545* 6609* 6610* 6611  6615* 6616  6621* 6624* 6630  6637  6644*
                  6648* 6662* 6670* 6674* 6689* 6697  6703* 6709* 6713* 6729* 6737  6770* 6771*
                  6772* 6773* 6774  6805  6810* 6811* 6813  6814  6850* 6961* 6993* 6994  6998*
                  7048* 7054* 7075  7077  7079  7105* 7112* 7114* 7115* 7116* 7117* 7118* 7119
                  7149* 7151* 7152* 7159  7274* 7278* 7284* 7288*
SPCHK  023316     5622*
SPLTST 026442     6399  6403*
SR0    = 177572   1149* 1372* 2261* 2274* 2315* 2509  6790  7047* 7160*
SR1    = 177574   1150* 6738  6801
SR2    = 177576   1151* 2381  6793
SR3    = 172516   1152* 7075* 7163*
SSP    =%000006   1083*
START  005666     1359  1385  2421*
START1 005776     1360  2447*
START2 006120     2466  2471* 6998  7156
START3 006050     1361  2306  2444  2459* 7194
STATUS 004023     2044  2105*
STKLIM 027410     6585*
STKPTR= 000500    1275* 2471  5062  5304  5457  5492  5506  6533  6644  6648
STMM   031622     6957  7004*
STORE  000764     1412*
SUB0   014736     4081  4082  4084*
SUB1   015352     4232  4233  4235*
SUB1A  015460     4271  4272  4274*
SUB1B  015474     4278  4279  4281*
SUB2   016170     4454  4455  4457*
SUB2A  016310     4496  4497  4499*
SUB3   016616     4604  4605  4607*
SUB3A  016640     4612  4613  4615*
SUB6   017254     4731  4733*
SUB7   017652     4828  4829  4830  4832*
SUCCES 004056     2112* 6977  7275
SWR0   010506     2897  2898  2899  2901*
SWR1   011714     3251  3252  3254*
```

```
SWR82   012120          3328    3330#
SWR84   012540          3486    3488#
SWR86   014006          3845    3847#
SWR87   014340          3952    3954#
SWR   = 177570          1153#   1453    1458    1460    1462    1472    1474    1494    1496    1498    1597    1692    2022
                        2025    2070    2073    2241    2299    6889    6929    6938    6966    6979    7010    7012    7016
                        7049    7098    7100    7102    7185
SW03  = 000010          1308#   1496    7100
SW04  = 000020          1307#
SW05  = 000040          1306#   1494    1597    7098
SW06  = 000100          1305#
SW09  = 001000          1304#   1474    7010    7049
SW12  = 010000          1303#   1472    7012    7016    7049
SXRA    024304          5868#   5877#
SXRB    024306          5869#   5895
SXT0    023752          5762    5763    5764    5765    5768#
SXT1    024036          5795#
SXT2    024322          5875#
SXT3    024336          5881#
SXT4    024104          5813    5815#
SXT5    024346          5886#
SXT6    024246          5850    5851    5852    5853    5855#
SXT6A   024276          5861    5864#
SXT7    024412          5903#
T     = 000020          1091#   6051    6060    6589
TBITVE= 000014          1109#   5562    5577#   5580#   5595#   6055#   6057#   6084#   6085#   6452#   6453#   6488#   6489#
TKB   = 177562          1146#   1996    2213
TKISR   003314          1336    1995#
TKS   = 177560          1145#   2211    2487#
TKVEC = 000060          1116#   1335
TPB   = 177566          1148#   1895#   6837#   6856#   7178#
TPS   = 177564          1147#   1893    2539    6832    6838    6840#   6843#   6848#   6853#   6891    6894#   6917#   6945#
                        6964    6978#   7176    7179
TPVEC = 000064          1117#   6834#   6835#   6896    6897#   6944#
TRAPVE= 000034          1115#   2481#   2482#   5356    5359#   5361#   5381
TRAP1   022132          5358    5371#
TRAP1C  022156          5370    5381#
TRKSEC  002306          1633    1658    1704#   1706#   1712#
TRTVEC= 000014          1110#
TSTB1   011726          3257    3258    3260#
TSTB2   012516          3475    3477#
TSTB2A  012526          3480    3482#
TSTB6   013534          3759    3760    3761    3763#
TST0    010244          2805    2806    2807    2808    2810#
TST1    011364          3124    3125    3126    3128#
TST2    012046          3302    3303    3305#
TST6    014036          3856    3857    3859#
TTOPT = 000400          1287#   1870    2541    6829    6962
TTYCHK  030466          5399    6827#
TYPADR  003302          1986#   2067    2148    2155    2270
TYPDAT  003270          1978#   2036    2046    2051    2054    2063    2135    2151    2158    2289    2293    2296    2549
                        2564    7280
TYPE  = 000004          1320#   1384    1604    1979    1987    2000    2007    2018    2032    2035    2041    2044    2049
                        2052    2061    2065    2072    2138    2145    2149    2152    2156    2217    2226    2231    2244
                        2269    2286    2287    2292    2294    2298    2308    2309    2403    2449    2453    2548    2550
                        2563    2565    6981    7187    7275    7281
```

```
UBM6    014042          3753  3755* 3758  3766  3770* 3777* 3779* 3785* 3793* 3800* 3806* 3813* 3819*
                        3825* 3831* 3837* 3844* 3849* 3855  3862*
UBREAK= 177770          1137* 1460*
UB7     017540          4790  4801*
UDPAR0= 177660          1271* 6753
UDPDR0= 177620          1268* 6745
UIPAR0= 177640          1251* 6503  6750  7060* 7061
UIPAR1= 177642          1252* 7061* 7062*
UIPAR4= 177650          1253* 6764*
UIPAR6= 177654          1254* 6503*
UIPAR7= 177656          1255* 7063*
UIPDR0= 177600          1246* 6742  7057*
UIPDR1= 177602          1247* 7058*
UIPDR4= 177610          1248* 6765*
UIPDR6= 177614          1249* 6504*
UIPDR7= 177616          1250* 7059*
UM    = 140000          1100* 5271  5432  6096  6102  6128  6417  6431  6493  6619  6627
USP   =%000006          1084*
UWM6    013230          3672* 3676* 3683* 3688* 3695* 3703* 3710* 3717* 3724* 3726* 3732* 3738* 3744*
UWM7    014132          3887*
UW7     014136          3885  3890*
V     = 000002          1088* 6185  6187  6213
VALDEV  002626          1576  1833* 2554  2557  2572  2574  2576  2578  2580  2582  2584  2607  2615
VIRPC   004015          2035  2103*
WAITIO  001370          1520* 1531  1607  1638  1643  1672
XOR0    024004          5776  5777  5778  5781*
XOR1    024072          5801  5802  5803  5804  5806  5809*
XOR24   024126          5823*
XOR35   024376          5892  5893  5897*
XOR6    024214          5828  5831  5833  5840*
XOR6A   024220          5826* 5827* 5832* 5834* 5838  5844* 5849* 5860* 5862*
XOR6B   024222          5835* 5836* 5837* 5838  5845*
XOR7    024426          5908*
Z     = 000004          1089* 5255
$FILLS  001002          1430* 1886
$RESTR  005410          1965  2069  2353*
$SAVF.  005370          1904  1924  2027  2343*
.     = 033170          1322* 1331* 1333* 1335* 1340* 1358* 1363* 1366* 1368* 1423* 1894  1970* 2023
                        2074  2126* 2267  2300  2339* 2392  2418* 2436  2635  2743  2755  2763  2771
                        2778  2786  2799  2809  2819  2827  2836  2845  2854  2863  2872  2878  2885
                        2892  2900  2922  2932  2942  2949  2956  2965  2981  3024  3034  3041  3050
                        3057  3063  3070  3077  3085  3092  3100  3105  3113  3119  3127  3134  3139
                        3144  3157  3163  3169  3175  3181  3187  3194  3201  3208  3216  3223  3227
                        3234  3239  3246  3253  3259  3263  3268  3274  3278  3284  3287  3292  3304
                        3309  3316  3323  3329  3334  3338  3346  3353  3360  3364  3370  3374  3380
                        3384  3387  3392  3406  3410  3414  3422  3430  3437  3445  3453  3459  3465
                        3471  3476  3481  3487  3495  3499  3502  3507  3520  3527  3533  3540  3548
                        3554  3560  3565  3571  3578  3584  3589  3608  3612  3619  3624  3632  3638
                        3643  3650  3654  3660  3666  3680  3686  3691  3699  3707  3713  3721  3729
                        3735  3741  3745  3748  3762  3767  3773  3782  3789  3796  3803  3809  3816
                        3822  3828  3834  3841  3846  3852  3858  3861  3880  3903  3910  3918  3925
                        3932  3940  3947  3953  3960  3964  3971  3976  3983  4000  4008  4014  4021
                        4027  4033  4039  4046  4052  4059  4066  4075  4083  4089  4098  4106  4126
                        4134  4144  4152  4160  4163  4173  4178  4187  4193  4208  4217  4225  4234
                        4241  4248  4254  4262  4273  4280  4285  4292  4300  4305  4310  4328  4332
                        4339  4350  4355  4361  4366  4372  4375  4424  4437  4444  4449  4456  4467
```

F13

		4477	4484	4490	4498	4505	4512	4545	4555	4559	4565	4572	4577	4606
		4614	4619	4622	4628	4633	4638	4670	4673	4679	4686	4692	4697	4710
		4715	4719	4726	4732	4739	4745	4749	4770	4774	4777	4782	4786	4814
		4817	4818	4824	4831	4839	4843	4847	4851	4881	4901	4910	4916	4921
		4925	4931	4936	4975	4980	4986	4990	5000	5010	5016	5052	5054*	5073
		5075	5080	5085	5093	5095	5100	5116	5120	5134	5142	5153	5236	5247
		5266	5291	5298	5310	5340	5341	5348	5358	5366	5368	5371	5375	5410
		5420	5494	5515	5566	5585	5630	5637	5645	5651	5662	5680	5695	5704
		5754	5767	5780	5791	5794	5808	5822	5839	5842	5854	5863	5867	5874
		5880	5885	5896	5902	5907	5919	5938	5955	5959	5961	5964	5989	5995
		6015	6020	6024	6054	6111	6126	6137	6164	6194	6217	6237	6261	6280
		6292	6311	6339	6365	6389	6415	6429	6467	6496	6512	6538	6545	6558
		6579	6610	6636	6773	6781	6811	6833	6839	6965	7180	7244*	7253*	7266*
.HLT	003416	1329	2022*	2246	2406	2481	5383							
.MAMF	000120	1345*	2462											
.PARSR	004622	1345	2241*	2303										
.TPVEC	031264	6896*	6943*	6944										
.TYPE	002736	1323	1867*	2448	2478	5264	6851							

```
COMMEN      1#
ENDCOM      1#
ESCAPE      1#
GETDAT   1322#   2205
GETPRI      1#
GETSWR      1#
KW11L    1322#
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
RELOCA   1322#   2625    2733    3870    4891    5400    5744    6154    6569
RELOCB   1322#   2727    3864    4885    5389    5737    6147    6563    6819
RELOCC   1322#   1470
REPORT      1#
RESTOR   1322#   1965    2068
SAVE     1321#   1903    1923    2027
SCOP4    1321#   1443
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
WHICHC   1322#   2491
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.ERRO    1322#   2020
.HEADE      1#
.KT11       1#
.RESTO   1322#   2351
.SAVE    1322#   2341
.SETUP      1#
.SWRHI      1#
.SYCT1      1#
.SYPTB      1#
.SYPTH      1#
.SYPTY      1#
.SYSTA      1#
.SCATC      1#
.SCMTA      1#
.SD82D      1#
.SD82O      1#
.SDIV       1#
.SEOP       1#
.SERRO      1#
.SERRT      1#
```

```
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
..TYPE   1322#   1865
.1170       1#
```

DCQKCG.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADC 1699 1703 1945 2823 3046 3104 3344 3568 3738 3968 4864 5676
ADCB 3153 3213 3409 3412 3641 3779 3793 4004 4064 5042 5320 5344
ADD 1487 1529 1629 1681 1682 1698 1702 1869 1937 1944 1961 2313 2382 2728 2953
3754 3865 4103 4169 4182 4205 4288 4297 4303 4434 4610 4626 4723 4741 4835
4862 4867 4874 4886 4968 4969 4970 4971 5022 5028 5058 5070 5090 5124 5146
5160 5176 5184 5187 5197 5220 5247 5266 5310 5358 5373 5390 5420 5425 5450
5464 5472 5504 5513 5518 5529 5599 5655 5667 5738 5859 6054 6062 6111 6126
6137 6148 6336 6419 6455 6459 6469 6472 6496 6509 6512 6538 6545 6564 6610
6636 6673 6686 6712 6725 6773 6789 6811 6820 7045 7062 7073 7091 7140
ASH 2514 6179 6205 6356 6360 6380 6384
ASHC 6230 6252 6287
ASL 1354 1575 1602 1620 1621 1622 1623 1696 1697 1932 1936 1940 2039 2233 2234
2235 2367 2368 2369 2506 2867 2909 2911 2913 2915 2960 2962 3008 3109 3373
3376 3537 3726 3915 4104 4114 4706 4707 4717 4729 4742 4816 5030 5469 5471
5473 5475 5477 5592 5677 5829 6183 6231 6295 6344 6460 6600 6843 6848 6918
6988 7107 7108 7109 7110 7111 7113 7115 7116 7117 7118
ASLB 3178 3265 3449 3635 3785 4036 4331 5293
ASR 1612 1950 2220 2221 2881 2946 2974 3066 3350 3531 3688 3724 3929 4834 6210
6255 6304 6787 6840
ASRB 3211 3219 3220 3434 3442 3616 3825 4043 4412 4422 5339
BCC 1355 2592 2603 2776 2815 2824 2841 2860 2868 2882 2890 2947 2954 2963 3040
3054 3067 3082 3110 3117 3132 3138 3167 3173 3179 3192 3199 3206 3221 3244
3272 3308 3314 3333 3351 3358 3364 3377 3404 3419 3450 3457 3469 3492 3538
3545 3552 3559 3564 3576 3607 3617 3623 3629 3636 3665 3684 3696 3704 3727
3733 3781 3786 3807 3814 3826 3832 3838 3850 3908 3916 3930 3945 3958 3981
3997 4012 4019 4025 4031 4037 4044 4057 4073 4119 4132 4141 4159 4170 4206
4223 4246 4252 4259 4271 4289 4323 4348 4354 4359 4435 4442 4448 4464 4475
4482 4570 4612 4632 4708 4724 4813 4828 4836 4998 5008 5077 5170 5215 5294
5335 5345 5593 5692 5762 5801 5931 5936 6601
BCS 1592 2510 2515 2520 2524 2528 2532 2536 2540 2570 2748 2795 2805 2832 2850
2876 2897 2930 3047 3074 3089 3097 3124 3161 3185 3215 3232 3251 3257 3266
3302 3328 3368 3413 3427 3435 3443 3463 3475 3486 3525 3569 3583 3642 3659
3677 3689 3711 3718 3739 3759 3771 3794 3801 3820 3845 3856 3923 3952 3969
4005 4050 4081 4088 4095 4214 4232 4298 4454 4496 4563 4604 4670 4731 4774
4822 5366 5612 5701 5776 5852 5946 6326 6337
BEQ 1451 1454 1459 1473 1475 1497 1506 1512 1523 1527 1598 1693 1871 1930 2038
2060 2071 2080 2085 2133 2373 2392 2427 2466 2573 2575 2577 2579 2581 2583
2585 2750 2817 2834 2843 2852 2962 2922 2932 2942 2949 2956 2965 2981 3076
3084 3091 3099 3139 3169 3181 3187 3208 3216 3227 3263 3278 3284 3287 3304
3322 3338 3379 3384 3387 3410 3421 3429 3452 3465 3476 3480 3494 3499 3502
3520 3547 3578 3584 3612 3660 3666 3679 3698 3706 3720 3748 3767 3773 3834
3840 3852 3858 3903 3983 3999 4007 4039 4052 4066 4083 4089 4106 4152 4163
4172 4178 4187 4208 4216 4225 4234 4261 4284 4305 4325 4328 4350 4360 4366
4372 4402 4405 4414 4424 4437 4456 4466 4477 4490 4505 4545 4555 4572 4577
4606 4619 4638 4673 4679 4692 4710 4715 4719 4732 4739 4745 4749 4770 4777
4782 4786 4818 4830 4838 4843 4851 4910 4916 4921 4925 4931 4936 4975 4980
4986 4990 5000 5010 5100 5116 5134 5142 5153 5178 5202 5236 5298 5316 5348
5368 5375 5494 5515 5520 5532 5566 5576 5579 5585 5590 5614 5630 5637 5645
5651 5662 5680 5694 5703 5764 5767 5778 5780 5794 5822 5839 5861 5863 5874
5880 5892 5896 5902 5907 5919 5938 5943 5952 5955 5964 5980 5989 5995 6020
6024 6070 6075 6098 6101 6118 6129 6141 6186 6194 6217 6237 6261 6286 6292
6311 6332 6339 6365 6389 6415 6429 6467 6494 6502 6506 6516 6558 6586 6596
6620 6639 6669 6672 6708 6711 6761 6767 6796 6857 6871 6876 6913 6930 6939
6952 6963 6995 7011 7017 7067 7101 7130 7137 7162 7189
BGE 2761 2771 2827 2863 2878 2900 3034 6436
BGT 2762 2778 2784 2836 2854

BHI	1489	2755	2777	2785											
BIC	1524	1701	1706	1857	1862	1931	1943	1997	2214	2232	4140	4149	4251	4463	4487
	4631	4713	4842	4880	5036	5254	5278	5302	5588	6066	6406	6408	6470	6649	6677
	6682	6716	6721	6788	6917	6945	6989								
BICB	1573	3005	4358	4369	4419	4423	4541	4576	4681	4682	4773	4780	4908	4989	5006
	5641	6213													
BIS	1452	1525	1572	1663	2236	2370	2487	2511	2516	2521	2525	2529	2533	2537	2541
	2587	2593	2598	2604	2609	2612	2617	2917	2918	2919	2920	4131	4158	4245	4345
	4447	4481	4609	4705	4768	4846	5059	5250	5255	5279	5284	5418	5439	5440	5449
	5478	5482	5485	5488	5573	5582	6058	6060	6123	6134	6425	6457	6461	6591	6849
	6853	6955	6978												
BISB	3000	4353	4400	4403	4536	4558	4676	4677	4914	4934	4978	4996	5040	5446	6187
	6190	6894													
BIT	1450	1453	1458	1462	1472	1474	1494	1496	1597	1692	1870	2025	2070	2596	2985
	4118	4125	4222	4474	4618	4714	4744	4861	5271	5378	5432	5575	5578	6051	6102
	6168	6331	6417	6431	6493	6585	6589	6619	6627	6829	6832	6870	6875	6889	6891
	6929	6938	6951	6962	6964	6994	7010	7012	7016	7049	7098	7100	7176		
BITB	4347	4401	4404	4413	4569	4683	4685	4769	4785	4866	4869	4877	4924	4974	5052
	5514	5648	6185	6887											
BLE	1619	2753	2770	2786	2799	2845	2877	2892	3041						
BLOS	2754	2809	2872	3032	3061	3648	3938	4278	5397						
BLT	1889	2752	2763	2769	2819	2885	6422								
BMI	2751	2798	2808	2844	2871	3077	3085	3092	3105	3119	3154	3163	3175	3234	3246
	3259	3268	3309	3323	3329	3334	3346	3353	3360	3380	3406	3422	3437	3453	3471
	3481	3487	3527	3533	3540	3554	3571	3608	3619	3638	3643	3654	3686	3691	3707
	3721	3729	3741	3745	3778	3796	3803	3809	3822	3828	3841	3846	3910	3918	3925
	3932	3947	3953	3960	3964	3976	4000	4014	4027	4033	4046	4059	4115	4126	4160
	4217	4248	4254	4273	4285	4300	4339	4355	4444	4449	4467	4484	4498	4559	4614
	4628	4633	4814	4824	4831	4847	5232	5272	5332	5371	5695	5804	5828	5851	5885
	5944	5953	5970	6327	6458	6980	7005								
BNE	1383	1447	1463	1465	1477	1484	1493	1495	1504	1510	1578	1594	1596	1650	1678
	1691	1873	1876	1884	1954	1959	1964	1999	2006	2012	2026	2048	2058	2078	2137
	2141	2216	2225	2279	[illegible]	2285	2441	2547	2558	2562	2597	2608	2616	2783	2797
	2807	2926	2870	2884	2899	2940	2977	2979	2986	3007	3009	3050	3056	3069	3112
	3127	3134	3194	3223	3414	3631	3761	3788	4075	4097	4121	4134	4143	4185	4291
	4370	4408	4411	4417	4420	4508	4528	4530	4533	4535	4538	4540	4543	4684	4686
	4690	4752	5079	5097	5113	5131	5165	5167	5172	5192	5199	5217	5222	5268	5270
	5273	5275	5281	5286	5296	5346	5433	5559	5628	5635	5643	5649	5658	5660	5803
	5806	5850	5928	5933	5949	6052	6068	6092	6103	6115	6169	6192	6198	6215	6221
	6235	6241	6259	6265	6290	6309	6330	6363	6371	6387	6395	6399	6418	6432	6524
	6590	6629	6631	6633	6664	6680	6685	6690	6705	6719	6724	6730	6781	6791	6794
	6799	6802	6806	6830	6833	6847	6888	6890	6892	6965	6985	7013	7050	7053	7097
	7099	7126	7135	7177	7183										
BPL	1371	1645	1674	1894	2023	2074	2212	2242	2250	2273	2300	2429	2760	2818	2835
	2853	3033	3049	3057	3063	3070	3100	3113	3126	3201	3253	3274	3316	3370	3430
	3445	3459	3495	3548	3565	3624	3632	3650	3680	3699	3713	3735	3762	3789	3816
	3940	3971	4008	4021	4098	4144	4173	4241	4262	4280	4292	4326	4361	4565	4726
	4839	5080	5173	5218	5323	5327	5337	5572	5704	5765	5791	5831	5833	5893	5929
	5934	6280	6659	6839	6895	6905	6967	7180	7186						
BPT	5474														
BR	1468	1600	1605	1666	1679	1864	1891	2015	2043	2064	2083	2222	2237	2267	2290
	2310	2383	2387	2436	2444	2567	2588	2594	2599	2605	2610	2613	2618	3010	3024
	3144	3156	3214	3292	3392	3507	3589	3780	3861	3885	4113	4193	4310	4375	4512
	4582	4643	4697	4790	4881	4941	5016	5056	5073	5075	5085	5093	5095	5106	5108
	5120	5126	5128	5137	5139	5147	5149	5168	5188	5193	5208	5209	5214	5228	5229
	5276	5282	5341	5370	5443	5452	5461	5479	5527	5534	5785	5813	5842	5854	5867

	5915	5924	5959	5968	6015	6018	6022	6072	6077	6079	6088	6120	6131	6333	6346
	6518	6543	6622	6691	6696	6731	6736	6739	6783	6842	6974	6915	6924	7276	7282
	7286	7294													
BVC	1501	2768	2833	2842	2851	2861	2869	2883	2961	3048	3055	3068	3075	3098	3111
	3162	3168	3180	3200	3212	3222	3233	3267	3315	3321	3369	3436	3458	3526	3532
	3546	3560	3570	3577	3630	3690	3712	3719	3734	3740	3787	3795	3802	3821	3851
	3975	3982	4020	4026	4038	4171	4272	4290	4299	4332	4338	4564	4613	4627	4725
	4817	4829	4837	5078	5216	5291	5295	5318	5322	5326	5331	5336	5678	5830	5930
	5935	6296	6328	6345	7121										
BVS	2749	2796	2806	2816	2825	2891	2898	3062	3063	3090	3118	3125	3133	3157	3174
	3186	3193	3207	3239	3245	3252	3258	3273	3303	3345	3352	3359	3374	3378	3405
	3420	3428	3444	3451	3464	3470	3493	3539	3553	3618	3637	3649	3678	3685	3697
	3705	3728	3760	3772	3782	3808	3815	3827	3833	3839	3857	3909	3917	3924	3931
	3939	3946	3959	3970	3998	4006	4013	4032	4045	4051	4058	4074	4082	4096	4120
	4133	4142	4207	4215	4224	4233	4247	4253	4260	4279	4324	4349	4436	4443	4455
	4465	4476	4483	4497	4571	4605	4622	4823	5171	5340	5693	5702	5763	5777	5802
	5808	5853	5945	6253	6285										
CCC	2747	2812	2848	2927	2972	3037	3044	3319	3433	3518	3530	3574	3765	3921	3979
	4092	4124	4336	4432	4625	4722	5224	5609	5878	5950	6412				
CFCC	2519														
CLC	1956	2218	3197	3204	3312	3402	3627	3914	3945	4017	4042	4229	5690	5774	
CLN	3122	3536	3950	4212	4462	5074	5564	5690	5872	5900					
CLR	1372	1380	1448	1491	1499	1514	1907	1928	1949	2076	2081	2210	2248	2252	2258
	2275	2405	2434	2460	2483	2484	2488	2497	2505	2512	2527	2555	2620	2794	2905
	2936	2969	3028	3150	3297	3337	3399	3512	3597	3671	3755	3893	3991	4111	4129
	4198	4200	4265	4315	4317	4342	4493	4598	4599	4703	4857	4858	4913	4959	4965
	4966	5083	5103	5174	5182	5249	5288	5308	5351	5384	5416	5441	5467	5468	5516
	5560	5577	5581	5594	5625	5759	5789	5890	5914	5917	5922	6059	6085	6108	6124
	6175	6228	6229	6249	6250	6273	6277	6278	6322	6350	6351	6353	6373	6374	6377
	6405	6447	6449	6484	6486	6498	6588	6602	6624	6634	6642	6647	6662	6665	6666
	6676	6688	6703	6706	6715	6728	6762	6763	6764	6765	6768	6769	6776	6827	6898
	6931	6940	6959	6991	7041	7048	7087	7105	7139	7150	7160	7163	7181	7289	
CLRB	2247	2425	2461	2486	3262	4237	4380	4382	4384	4386	4993	5451	5664	6095	6487
	6530	6910	7171	7178											
CLRD	5050														
CLRF	5034														
CLV	1684	2839	2858	3159	3237	3716	3743	3792	5162	5328	5333	5338	5774		
CLZ	2793	2803	3300	3757	3901	4086	4139	4221	4229	4452	4575	4602	5587		
CMP	1488	1503	1505	1509	1526	1998	2132	2278	2282	2557	2572	2574	2576	2578	2580
	2582	2584	2921	3006	3286	3383	3386	3498	3501	3674	3747	4094	4105	4151	4176
	4177	4184	4186	4213	4258	4441	4489	4507	4527	4529	4532	4534	4537	4539	4542
	4544	4589	4591	4621	4637	4652	4672	4675	4678	4709	4718	4748	4751	4802	4804
	4807	4821	4850	4865	4868	4876	4878	4909	4915	4920	4930	4950	4999	5099	5109
	5114	5115	5133	5166	5177	5198	5201	5210	5221	5234	5235	5267	5269	5274	5280
	5285	5374	5437	5493	5531	5584	5627	5629	5636	5642	5644	5650	5661	5679	5691
	5779	5784	5793	5812	5838	5895	5906	5948	6019	6023	6067	6069	6117	6128	6140
	6191	6197	6214	6220	6234	6236	6240	6242	6258	6260	6264	6266	6289	6291	6308
	6310	6329	6338	6362	6370	6385	6394	6421	6435	6466	6557	6595	6632	6637	6679
	6684	6718	6723	6790	6793	6805	6912	6984	7125	7129	7134	7159			
CMPB	1446	1618	1883	2005	2011	2034	2047	2056	2215	2224	4337	4365	4407	4410	4416
	4504	4531	4562	4689	4691	4776	4781	4860	4985	5377	5396	5657	5659	5700	6091
	6193	6216	6364	6388	6398	6414	6428	6505	6663	6704	6766	6795	6798	6801	6846
	7052	7066	7161												
COM	2013	2280	2814	2941	2970	3116	3307	3558	3683	3957	4148	4150	4156	4166	4202
	4230	4269	4295	4460	4472	4488	4494	4624	4636	4743	4747	5094	5107	5127	5138
	5148	5194	5640	5773	5792	5820	5837	5894	6080	6281	6302	6305	6525	6681	6720

COMB	3205	3271	3403	3606	3806	4011	4065	4319	4364	4928	5319				
DEC	1953	1958	1963	2077	2140	2840	2939	2976	3039	3137	3367	3544	3732	3922	4521
	5947	6219	6263	6392	6393	6434	6689	6729							
DECB	1464	1649	1677	1888	3191	3226	3238	3243	3456	3664	3837	4030	4334	4418	4983
	5320														
DIV	6324														
EMT	1319	5315	5355	6514											
FADD	2523														
HALT	1322	1374	2002	2024	2075	2243	2301	2393	6541						
INC	1382	2136	2274	2401	2551	2556	2566	2849	2931	2948	2978	2980	2993	3081	3149
	3320	3398	3563	3581	3601	3717	3974	3989	4112	4162	4203	4318	4657	4662	4762
	4953	4957	4984	5020	5096	5112	5130	5141	5152	5191	5533	5613	5688	5766	5862
	5969	5992	6064	6196	6239	6294	6299	6368	6369	6420	6515	6667	6854	6858	6921
	6927	6935	6982	7292											
INCB	1513	1599	3155	3160	3281	3440	3491	3622	3777	3831	4024	4062	4063	4327	4406
	4935	4979	5334	5470	5633	7138									
IOT	1320	5258													
JMP	1385	1530	1638	1651	1685	1909	2086	2090	2246	2306	2406	2621	2639	5072	5092
	5111	5129	5140	5150	5399	6093	6170	6400	6831	6986	7014	7018	7156	7172	7194
JSR	1500	1507	1570	1581	1631	1683	1877	1882	1890	1904	1924	1965	1978	1986	202[illegible]
	2029	2036	2042	2046	2051	2054	2063	2067	2069	2135	2148	2151	2155	2158	2[illegible]70
	2289	2293	2296	2450	2454	2462	2485	2549	2564	2632	2740	3877	4898	5163	5190
	5212	5231	5407	5417	5476	5561	5605	5751	6014	6161	6576	6851	6957	6960	6969
	7030	7080	7120	7131	7190	7280									
MARK	6012														
MFPD	5032	5048													
MFPI	6520	6521	6522												
MOV	1345	1346	1347	1348	1349	1351	1359	1360	1361	1373	1377	1379	1449	1455	1456
	1467	1478	1479	1480	1482	1485	1486	1502	1515	1516	1520	1571	1576	1577	1582
	1587	1590	1603	1607	1609	1611	1614	1616	1624	1625	1626	1627	1628	1630	1632
	1633	1634	1635	1636	1637	1640	1643	1644	1647	1656	1657	1658	1659	1660	1661
	1662	1664	1669	1672	1673	1676	1680	1694	1695	1700	1704	1707	1709	1711	1[illegible]
	1844	1845	1846	1847	1848	1849	1850	1854	1855	1858	1863	1867	1868	1879	1886
	1905	1906	1908	1925	1927	1938	1939	1946	1948	1955	1996	2028	2030	2031	2033
	2037	2040	2045	2050	2053	2055	2066	2088	2130	2131	2134	2139	2142	2146	2150
	2153	2157	2209	2251	2253	2254	2255	2256	2257	2259	2260	2261	2262	2263	2264
	2276	2277	2281	2288	2295	2303	2304	2305	2314	2315	2343	2344	2345	2346	2347
	2348	2349	2353	2354	2355	2356	2357	2358	2359	2363	2365	2374	2375	2380	2381
	2386	2390	2400	2422	2432	2433	2438	2443	2447	2448	2452	2456	2459	2463	246[illegible]
	2467	2471	2472	2473	2474	2475	2476	2477	2478	2479	2480	2481	2482	2493	2[illegible]
	2495	2507	2542	2543	2544	2553	2554	2559	2628	2630	2631	2633	2634	2636	26[illegible]
	2727	2729	2736	2738	2739	2741	2742	2744	2745	2908	2910	2912	2914	2916	29[illegible]
	2945	2959	2971	2975	2987	2988	2990	2994	2995	2996	2997	2998	2999	3001	300[illegible]
	3004	3011	3012	3013	3014	3015	3016	3019	3020	3026	3146	3148	3294	3296	33[illegible]
	3396	3397	3510	3513	3515	3516	3593	3596	3600	3602	3603	3604	3673	3753	3864
	3866	3873	3875	3876	3878	3879	3881	3882	3890	3894	3895	3898	3899	3992	4072
	4078	4087	4101	4102	4116	4123	4137	4147	4155	4157	4165	4175	4181	4183	4196
	4199	4240	4268	4283	4296	4313	4316	4378	4381	4383	4385	4429	4430	4431	4470
	4471	4502	4503	4515	4519	4520	4522	4523	4549	4552	4587	4588	4590	4592	4593
	4594	4596	4597	4651	4653	4654	4656	4658	4659	4661	4663	4666	4701	4702	4704
	4712	4736	4737	4759	4760	4761	4763	4765	4767	4801	4803	4805	4806	4808	4811
	4812	4859	4885	4887	4894	4896	4897	4899	4900	4902	4903	4906	4907	4949	4951
	4952	4955	4956	4960	4961	4962	4967	5014	5015	5019	5021	5023	5062	5063	5064
	5069	5084	5088	5089	5091	5104	5105	5119	5123	5125	5145	5158	5159	5175	5181
	5183	5185	5186	5196	5205	5206	5207	5219	5230	5245	5246	5248	5252	5253	5256
	5257	5261	5263	5264	5265	5303	5304	5309	5311	5313	5350	5356	5357	5359	5361

```
        5363  5381  5382  5383  5385  5389  5391  5395  5398  5403  5405  5406  5408  5409  5411
        5412  5415  5419  5421  5422  5423  5426  5427  5429  5430  5431  5438  5442  5447  5448
        5455  5456  5457  5465  5492  5498  5499  5500  5501  5503  5506  5507  5508  5512  5517
        5519  5521  5522  5528  5552  5559  5562  5563  5569  5574  5580  5583  5589  5595  5596
        5598  5603  5608  5615  5616  5619  5622  5623  5654  5656  5665  5666  5672  5674  5686
        5687  5737  5739  5747  5749  5750  5752  5753  5755  5756  5770  5771  5772  5783  5797
        5798  5811  5815  5818  5825  5835  5848  5857  5858  5866  5871  5877  5888  5926  5941
        5942  5958  5967  5977  6003  6004  6005  6006  6007  6008  6009  6010  6011  6012  6013
        6026  6050  6053  6055  6056  6057  6061  6078  6081  6083  6084  6094  6096  6099  6106
        6107  6109  6110  6113  6125  6135  6136  6144  6147  6149  6157  6159  6160  6162  6163
        6165  6166  6173  6174  6176  6177  6178  6200  6201  6202  6203  6204  6223  6224  6225
        6226  6227  6244  6245  6246  6247  6248  6272  6274  6275  6279  6298  6300  6303  6319
        6320  6321  6334  6352  6354  6355  6357  6378  6379  6381  6403  6407  6410  6411  6423
        6444  6446  6448  6450  6452  6453  6454  6471  6485  6488  6489  6495  6497  6499  6500
        6503  6504  6507  6508  6511  6513  6531  6532  6533  6537  6539  6540  6544  6548  6549
        6554  6560  6563  6565  6572  6574  6575  6577  6578  6580  6581  6587  6593  6594  6609
        6611  6612  6613  6615  6616  6617  6621  6630  6635  6643  6644  6646  6648  6650  6651
        6660  6661  6668  6670  6674  6675  6678  6683  6701  6702  6707  6709  6713  6714  6717
        6722  6770  6771  6772  6774  6775  6777  6778  6779  6785  6792  6797  6800  6803  6810
        6813  6814  6819  6821  6828  6834  6835  6836  6850  6865  6866  6867  6868  6869  6872
        6873  6878  6879  6880  6896  6897  6908  6909  6922  6923  6928  6936  6937  6944  6946
        6947  6953  6954  6958  6961  6968  6970  6971  6972  6973  6974  6975  6976  6977  6987
        6990  6993  6997  6998  7027  7029  7033  7034  7035  7036  7037  7038  7039  7042  7043
        7044  7046  7047  7054  7057  7058  7059  7060  7061  7063  7068  7069  7070  7071  7072
        7074  7075  7081  7082  7086  7089  7090  7092  7093  7106  7122  7123  7127  7128  7142
        7143  7149  7151  7152  7170  7188  7274  7278  7279  7284  7288  7290  7291
MOVB    1460  1466  1498  1574  1601  1608  1615  1654  1875  1895  1962  2014  2017  2213  2230
        2245  2371  2430  2928  2952  4322  4393  4395  4397  4421  4526  4550  4551  4553  4554
        4667  4668  4669  4735  4766  4919  4929  4973  5004  5466  5568  5604  5610  5626  6180
        6184  6188  6206  6212  6361  6385  6451  6456  6837  6856  6893  6903  6983  6992  6996
        7079  7102  7119
MTPI    6526
MUL     6283  6306  6335
NEG     1913  2284  2859  2895  3131  3332  3551  3695  3944  4459  4617  5807  6209  6254  6375
NEGB    3184  3468  3663  3813  4049  5054  5325
NOP     1665  1995  4870  4875  5711  5712  5713  5714  5715  5716  5717  5718  5719  5720  5721
        5722  5723  5724  5725  5726  5727  5728  5729  5730  5731  5732  5733  5734  6462  6510
        6519  6523  6880  6900  6901  6902  6906  7078  7155  7164  7191  7192  7193
RESET   2271  2302  2421  6556  7184
ROL     1933  1934  1935  1941  2875  2907  2929  3053  3060  3357  3575  3676  3980  6232  7112
        7114
ROLB    3166  3426  3647  3819  3849  4056  4392  4394  4396  4398  4399  4409  4997  5290
ROR     1951  2219  2831  2938  2973  3031  3073  3313  3342  3524  3703  3937  4168  4267  4344
        5611  5923  6256
RORB    1952  1957  3198  3231  3418  3628  3800  4018  4333  4335  4415  5007  5321
RTI     1365  1461  1641  1670  1859  1880  2003  2009  2019  2087  2089  2316  2433  2474  5060
        5343  5379  5398  5483  5485  5505  5508  5530  5580  5600  5668  6078  6082  6127  6528
        6546  6651  6807  6812  6844  6855  6860  6919  6933  6941  7153
RTS     1356  1584  1708  1713  1851  1860  1896  1966  1980  1988  2143  2159  2228  2360  2376
        5169  5195  5224  5239  5486  6016  7006
RTT     6033  6067  6083  6112  6138  6473  6999
SBC     2498  2501  2504  2889  3088  3096  3363  3582  3710  3907
SBCB    3172  3283  3462  3658  3770  3996
SCC     2792  2802  2838  2857  2888  2906  3190  3249  3299  3401  3425  3441  3490  3543  3646
        3675  3694  3702  3756  3900  3906  3913  3967  4003  4071  4117  4130  4138  4211  4220
        4228  4276  4321  4451  4461  4601  5025  5071  5161  5211  5689  5699  5760  5799  5927
```

```
        6282  6323  6426  6553  6555
SEC     1588  2496  2499  2502  2508  2513  2517  2522  2526  2530  2534  2538  2568  2590  2601
        2775  2822  2830  2866  2937  3030  3038  3045  3080  3095  3103  3108  3152  3230  3242
        3282  3326  3341  3343  3417  3484  3523  3657  3725  3776  3799  3995  4167  4244  4266
        4343  4346  4390  4568  4630  4665  4764  4810  5289  5314  5329  5362
SEN     2759  4093  4239  4346  5238  5360  5883
SEV     1583  2767  2813  3130  3349  3356  3417  3448  3557  3615  3812  3928  3936  3943  3956
        3995  4055  4079  4093  4204  4244  4257  4346  4433  4440  4473  4480  5312  6251
SEZ     2782  2813  3448  5314  5364
SOB     1942  5932  5951  5961  5974  5978  5993  6189  6211  6233  6257  6687  6726
SPL     6404  6410  6411  6413  6424  6427
SUB     1457  1481  1490  1511  1648  1655  1926  2062  2437  2442  2635  2743  3027  3147  3295
        3395  3511  3880  4080  4231  4270  4277  4453  4495  4603  4611  4730  4738  4827  4873
        4879  4901  5026  5038  5044  5410  5675  5754  5836  6164  6433  6579  7136  7182
SWAB    2896  3250  3277  3327  3485  3653  3744  3844  3951  3963  4238  4243  5003  6786
SXT     5761  5788  5816  5826  5849  5860  5873  5879  5884  5901
TRAP    1318  5365  5387
TST     1370  1483  1589  1593  1705  1872  1878  2001  2008  2022  2059  2073  2079  2082  2147
        2154  2227  2241  2249  2266  2272  2299  2391  2426  2428  2435  2500  2509  2531  2535
        2539  2545  2560  2591  2602  2607  2615  2629  2737  2804  2955  2964  3123  3301  3514
        3519  3594  3595  3598  3599  3855  3874  3891  3892  3896  3897  3902  3988  3990  4197
        4304  4314  4379  4516  4524  4595  4655  4660  4895  4954  5009  5018  5046  5087  5098
        5110  5122  5132  5164  5200  5223  5297  5347  5404  5436  5565  5571  5634  5748  5787
        5790  5805  5819  5870  5899  5918  5937  5954  5963  5979  5983  5994  6073  6114  6158
        6573  6658  6697  6737  6780  6804  7004  7077
TSTB    1476  1492  1522  1595  1690  1893  1929  2057  2084  2211  2372  2440  2465  2503  2569
        3256  3474  3479  3611  3758  3766  4371  4863  5557  6501  6671  6710  6760  6838  6966
        6979  7096  7179  7185
TSTF    2518
WAIT    6841
XOR     5775  5800  5817  5821  5827  5832  5834  5889  5891  5905  6097  6100
.ABS    1023
.ASCII  1403  2097  2112  2113  2162  2174  2175  2176  2177  2178  2179  2180  2181  2182  7245
        7258
.ASCIZ  1387  1390  1405  1406  2102  2103  2105  2106  2107  2108  2109  2110  2125  2167  2169
        2170  2318  2322  2328  2331  2335  2337  2408  2411  2414  6861  7235  7238  7241  7246
        7252  7264  7268
.BLKB   1970
.BYTE   1407  1408  1409  1421  1422  1424  1425  1969  1971  1972
.ENABL     1
.END    7297
.EVEN   1410  2111  2126  2172  2339  2418  6862  7244  7253  7266  7270
.LIST      1  1022  1322
.MACR   1321  1322
.MACRO     1  1321  1322
.NLIST     1  1021  1322
.PAGE   1322  1393  1470  1534  1865  2020  2239  2340  2378  2625  6957  7002
.REM       1
.REPT   1322  2640  5711
.SBTTL  1025  1056  1339  1393  1470  1534  1733  1865  2020  2239  2340  2378  2420  2626  2734
        3871  4892  5401  5745  6155  6570  6825  7002  7254
.TITLE  1024
.WORD   1323  1324  1325  1326  1327  1328  1329  1330  1332  1334  1336  1337  1364  1365  1367
        1381  1398  1399  1400  1401  1402  1411  1412  1413  1420  1426  1427  1428  1429  1430
        1436  1438  1440  1441  1469  1521  1531  1610  1617  1687  1710  1712  1716  1717  1718
        1719  1720  1721  1722  1723  1724  1725  1726  1727  1728  1729  1730  1731  1734  1735
```

1736	1737	1738	1739	1740	1741	1743	1744	1745	1746	1747	1748	1749	1750	1751
1753	1754	1755	1756	1757	1758	1759	1760	1761	1763	1764	1765	1766	1767	1768
1769	1770	1771	1773	1774	1775	1776	1777	1778	1779	1780	1781	1784	1785	1786
1787	1788	1789	1790	1791	1792	1793	1794	1795	1796	1797	1800	1801	1802	1803
1804	1805	1806	1807	1808	1809	1810	1811	1812	1813	1816	1817	1818	1819	1820
1821	1822	1823	1824	1825	1826	1827	1828	1829	1833	1834	1835	1836	1837	1838
1839	1840	1856	2185	2186	2187	2188	2189	2190	2191	2192	2193	2195	2196	2197
2198	2199	2200	2201	2202	2203	2364	2366	2402	2404	2439	2451	2455	2489	2546
2561	2640	2643	2646	2649	2652	2655	2658	2661	2664	2667	2670	2673	2676	2679
2682	2685	2688	2691	2694	2697	2700	2703	2706	2709	2712	2715	2718	2721	2724
2730	2989	2991	3003	3025	3145	3293	3393	3508	3509	3590	3591	3592	3672	3862
3867	3886	3887	3888	4194	4195	4311	4312	4376	4377	4513	4514	4583	4584	4585
4586	4644	4645	4646	4647	4648	4649	4698	4699	4791	4792	4796	4797	4798	4799
4888	4942	4943	4944	4945	4946	4947	5017	5086	5121	5151	5189	5392	5424	5502
5523	5597	5673	5740	5786	5814	5844	5845	5868	5869	6089	6150	6181	6207	6276
6301	6347	6358	6382	6566	6741	6742	6743	6744	6745	6746	6747	6749	6750	6751
6752	6753	6754	6755	6822	6859	6899	6904	6943	7028	7141	7228	7229	7230	7231
7232	7255													

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DCQKCG.SEQ/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZ=SYSMAC.CO,DCQKCG.P11
RUN-TIME: 44 68 11 SECONDS
RUN-TIME RATIO: 469/124=3.7
CORE USED: 41K (81 PAGES)

Spooler runtime 29 Seconds. 00 MPS, 574 disk reads, 0 disk writes, 170 pages